THE DIRECT DETERMINATION OF NONLINEAR DISPLACEMENTS OF ARBITRARILY SUPPORTED SHALLOW SHELLS USING MATHEMATICAL PROGRAMING TECHNIQUES

By

Harry G. Schaeffer

Thesis submitted to the Graduate Faculty of the
Virginia Polytechnic Institute
in candidacy for the degree of

DOCTOR OF PHILOSOPHY

in

ENGINEERING MECHANICS

April 1967

Blacksburg, Virginia

THE DIRECT DETERMINATION OF NONLINEAR DISPLACEMENTS OF ARBITRARILY SUPPORTED SHALLOW SHELLS USING MATHEMATICAL PROGRAMING TECHNIQUES

by

Harry G. Schaeffer

Thesis submitted to the Graduate Faculty of the

Virginia Polytechnic Institute

in candidacy for the degree of

DOCTOR OF PHILOSOPHY

in

ENGINEERING MECHANICS

APPROVED: Daniel Frederick

Chairman, D. Frederick

F. J. Maher

F. J. Maher

S T Gormson

S. T. Gormson

J. H. Sword

J. H. Sword

C W Smith

C. W. Smith

April 1967

Blacksburg, Virginia

THE DIRECT DETERMINATION OF NONLINEAR DISPLACEMENTS OF ARBITRARILY SUPPORTED SHALLOW SHELLS USING MATHEMATICAL PROGRAMING TECHNIQUES

By

Harry G. Schaeffer

ABSTRACT

The nonlinear behavior of an axisymmetrically loaded shallow shell of revolution with arbitrary edge conditions is investigated. The approach used in the present investigation is to obtain the displacement set which minimizes the finite-difference approximation of the shell potential energy by using mathematical programing techniques.

In order to determine the mathematical programing technique which is most suitable for minimizing the shell potential several methods which have appeared in the literature are evaluated. These include the method of steepest descent, the conjugate gradient method, the variable metric method and the generalized Newton-Raphson procedure. A combination of the conjugate gradient method and the generalized Newton-Raphson procedure is found most suitable.

Results of the present investigation are compared with published data for the buckling of a uniformly loaded clamped spherical cap. Additional results are presented which show the effect of changes in edge restraint and changes in shell geometry on the buckling pressure for uniformly loaded shallow shells.

II. TABLE OF CONTENTS

CHAPTER PAGE

III. LIST OF FIGURES AND TABLE

FIGURE PAGE

FIGURE PAGE

TABLE PAGE

IV. INTRODUCTION

In the last decade, significant progress has been made in two apparently unrelated fields as a result of the rapid development of efficient digital computers. These fields are mathematical programing and shell analysis.

The problem of mathematical programing is that of minimizing or maximizing an objective function $f\left(x_1, x_2, \ldots, x_n\right)$ by choice of the vector x_i. The problem of shell analysis is: given a set of equilibrium equations

$$Lx + Nx = b \tag{1}$$

where L is a linear operator and N is a nonlinear operator and b represents the nonhomogeneous terms, find the displacement set x which satisfies equation (1). Approximate solutions of the set of equation (1) have been obtained by solving a sequence of linearized sets of equations

$$(L + \overline{N})x = b \tag{2}$$

where $\overline{N}$ is the linear part of the operator N. For a shell having conservative internal and external forces a potential function exists, and the equilibrium equation (1) follow as a consequence of the Theorem of Minimum Potential through the application of the calculus of variations.

Approximate numerical solutions of the differential equations (1) or (2) are obtained by replacing the continuous displacement functions with a set of discrete values and by replacing differential operators with finite-difference operators. Equation (2) then becomes a set of linear algebraic equations where the operator $(L + \overline{N})$ is a square matrix of order n, where n is number of unknowns. The solution of the resulting set of equations can then be found by efficient matrix methods. It should be noted that solutions of equation (1) satisfy only a necessary condition for an extremum of the potential function. In order that the solution of the Euler-Lagrange equation correspond to a minimum of the potential, it is necessary and sufficient that all possible virtual displacements from the solution of the Euler-Lagrange equations lead to an increase in the potential.

All of the numerical investigations of shell behavior which have been reported in the literature seek solutions to the Euler-Lagrange equations. This is called the indirect method of solution. Another method of obtaining the solution is called the direct method wherein the displacement vector associated with an equilibrium state is found by the direct minimization of the finite-difference approximation of the potential function. The direct determination of the displacement set associated with the minimum of the potential function is a problem of mathematical programing. It should be noted that the displacement set which is determined by the direct method satisfies both sufficient and necessary conditions for a minimum.

The purpose of this investigation is to determine the feasibility of obtaining numerical solutions for nonlinear problems in continuum mechanics by minimizing a suitable function. The particular problem chosen for this investigation is that of determining the nonlinear displacements and the buckling load for an axisymmetrically loaded shallow shell of revolution having arbitrary edge restraints (linear springs). This problem was chosen because a significant literature exists for checking the nonlinear behavior of a uniformly loaded shallow spherical shell. However, the buckling behavior of shallow shells with edge conditions other than clamped and for shell geometry other than spherical has received little attention. A significant contribution to the literature of nonlinear shell analysis can therefore be made in these areas.

The approach to be investigated in the present study is that of obtaining the displacement set which minimizes the finite-difference approximation of the shell potential energy by using mathematical programing techniques. A thorough evaluation of the mathematical programing techniques which are presented in the literature is therefore required in order to determine the most suitable method of minimizing a function of a large number of variables.

This thesis represents a feasibility study of the direct method of solution, and while only a restricted number of variables are considered in the present investigation it is intended that the algorithm developed as a result of this study will be applied to problems in continuum mechanics which require a large number of variables. Therefore, numerical methods which are efficient with respect to both machine storage and machine execution time are desired.

Several algorithms for minimizing an object function were evaluated during the course of this investigation. A mathematical description of each algorithm is presented in chapter VII. The object function associated with the nonlinear shallow shell is derived in chapter VIII. A critique of the mathematical programing methods which were presented in chapter VII is given in chapter IX, and results for the shallow shell for various edge conditions are presented in chapter X. Also studied in chapter X is the effect of slight shape changes. The computer code prepared in the course of this investigation is presented in the appendix.

V. LIST OF SYMBOLS

a	radius of curvature of spherical segment or axis of ellipse
a^{α}	constant (see eq. (5))
b	normal radius to edge of shell
c^{α}	coefficient (see eq. (40))
c	axis of ellipse
d	bending stiffness parameter, $\frac{D}{D_o l^2}$
D	bending stiffness, $\frac{Et^3}{12(1-\nu^2)}$
D_o	reference bending stiffness
e_r, e_θ, $e_{r\theta}$	middle surface strains
E	Young's modulus of elasticity
E_o	reference modulus of elasticity
est	estimated minimum value of the object function, f(x)
f(x)	object function
H	shell height
k	membrane stiffness parameter, $12 \frac{E}{E_o} \left(\frac{b}{t_o}\right)^2 \left(\frac{t}{t_o}\right)$
k_r, k_θ	middle surface bending distortions
K	membrane stiffness, $\frac{Et}{(1-\nu^2)}$
$\overline{l}$	reference length
$l = \overline{l}/a$	

L	linear operator
$m^4 = 12\left(1-\nu^2\right)$	
n	number of unknowns
N	nonlinear operator
$\bar{N}$	linear part of nonlinear operator
NSTA	number of real shell stations
$p(x)$	surface pressure
$p_{cr} = 4E_o t_o{}^2 / a^2 m^2$	
$\bar{P} = p(x)/p_{cr}$	
q_1, q_2, q_3, q_4	boundary restraint coefficients (spring rates)
r	normal radius of shell middle surface
R_r, R_θ	principle radii of curvature of shell middle surface
s	tentative length of step in one-dimensional search
$\bar{s}$	length of step in which minimum of object function is bounded
t	shell thickness
t_o	reference shell thickness
$t^{\alpha\beta}$	coefficient (see eq.(27))
$\bar{u}$	displacement along shell meridian
u	nondimensional meridinal displacement, $\bar{u}/\bar{l}$
U	strain energy of shell
U_B	portion of strain energy associated with bending action

U_M	portion of strain energy associated with membrane action
$\overline{w}$	displacement normal to shell middle surface
w	nondimensional displacement, $\overline{w}/\overline{l}$
$y = \left(\tau^2 - f'(0)\ f'(\overline{s})\right)^{\frac{1}{2}}$	
z	vertical coordinate
β^{α}	coefficient, (see eq.(33))
$\gamma_{r\theta}$	shear strain
δ	convergence test parameter used in Newton-Raphson procedure
Δ	interval between shell stations, b/NSTA
$\Delta\Omega$	surface load increment
ϵ	orthogonality condition convergence test parameter
ϵ_r, ϵ_θ	strains
ϵ_i, $\epsilon_{i\pm 1/2}$	integration factors evaluated at station i, and midway between station i and $i\pm 1$, respectively
$\eta = c/a$	
$\zeta = z/b$	
θ	circumferential coordinate
λ	parameter denoting length of line in p_i-direction
λ_m	value of λ for which $f(x)$ is minimum

μ^2	geometric parameter for spherical shell, $\frac{m^2 b^2}{ah}$
ν	Poisson's ratio
$\bar{\Pi}_e$, $\bar{\Pi}_s$, $\bar{\Pi}_t$	potential energy of edge forces, surface forces, and total potential, respectively
$\bar{\Pi}_L$, $\bar{\Pi}_{NL}$	portions of $\bar{\Pi}_t$ giving rise to linear and nonlinear components of gradient, respectively
ρ	nondimensional radial coordinate, r/b
ρ_i, $\rho_{i+1/2}$	nondimensional radial coordinate evaluated at station i, and midway between station i and i+1, respectively

$$\tau = \frac{3(f(0) - f(\bar{s}))}{\bar{s}} + \frac{df}{d\lambda}(0) + \frac{df}{d\lambda}(\bar{s})$$

ω_1, ω_2	nondimensional curvatures, b/R_r and b/R_θ, respectively
Ω	nondimensional surface pressure $\frac{4}{l^2}\left(\frac{b}{a}\right)^2\left(\frac{b}{t_o}\right)\bar{P}$

Cartesian tensor quantities ($i, j, k, l = 1, 2, . . ., n$)

a_{ij}, $\bar{a}_{ij}$	coefficients of quadratic terms of f(x)
A_{ij}	defined by equation (60)
b_i	coefficients of linear terms of f(x)
B_{ij}	defined by equation (61)
c_{ijk}	coefficients of cubic terms of f(x)
d_{ijkl}	coefficients of quartic terms of f(x)

e_i	error vector, $\bar{h}_i - x_i$
g_i	gradient of $f(x)$
h_i	location of minimum of $f(x)$
$\bar{h}_i$	assumed location of minimum of $f(x)$
H_{ij}	variable metric
t_i	defined by equation (57)
p_i	conjugate gradient of $f(x)$
r_i	residual vector, see equation (18)
x_i	vector defining point on the surface $f(x)$ in Euclidian n-dimensional space
δ_{ij}	Kronecker delta, $= 0$ for $i \neq j$; $= 1$ for $i = j$

Subscripts:

In addition to subscripts on Cartesian tensor quantities the following subscripts are used:

b	quantity evaluated at shell edge
r, θ	denote radial and circumferential components

Superscript:

A Greek superscript is used to denote the iteration number in the conjugate gradient search procedure. A Roman superscript is used to denote the iteration number in the Newton-Raphson procedure.

Notation:

A prime indicates differentiation with respect to the independent variable.

VI. LITERATURE REVIEW

Direct Method of Solving Problems in Structural Mechanics

Rayleigh-Ritz Method.- The notion that a continuous system might be reduced to a system with a finite number of degrees of freedom by assuming a deflection shape is presented by Rayleigh (ref. 1). This concept was applied by Rayleigh in the study of vibrations of continuous media. The same concept has been used by Timoshenko (ref 2) in the study of the buckling of plates.

Rayleigh's method was generalized by Ritz (ref. 3) who considered the problem of determining an approximation of the function $y(x)$ which minimizes the integral

$$U = \int_a^b f(x,y,y',y'', \ . \ . \ .) \ dx$$

where f is a given function. The function y is restricted to a class which satisfy the geometric boundary conditions. For problems in continuum mechanics the function U may represent the total potential energy, for example. In the method presented by Ritz the solution $y(x)$ is taken in the following form

$$y \cong a_i g_i(x) \qquad (i = 1, \ . \ . \ . \ , n)$$

where the a_i are undetermined coefficients and the $g_i(x)$ are given functions each of which satisfies the geometric boundary conditions. The functions $g_i(x)$ are required to form a complete set in the interval (a,b) in order that the sequence converge in the mean square sense. The coefficients a_i are found from the necessary condition for the minimum of U,

$$\frac{\partial U}{\partial a_i} = 0 \qquad (i = 1, \ . \ . \ . \ , n)$$

The Ritz method has found wide use as a means of obtaining a solution by direct minimization of the potential function. For example,the Ritz method is used by Timoshenko in references 2 and 4 in the study of vibrations and plate deflection, respectively. Ketter (ref. 5) uses the Rayleigh-Ritz method to derive the flutter equations for a flat plate and Huang (ref. 6) uses the method to study the vibration of plates including rotary inertia and shear effects.

Numerical Solutions.- For problems where it is impractical to obtain the exact solutions, the variational methods exemplified by the Rayleigh-Ritz technique are popular and powerful. The main difficulty comes in choosing the appropriate complete set of functions each element of which satisfies the geometric boundary conditions.

Another method of solving problems which are not amenable to exact solution is the use of difference equations and appropriate numerical techniques. The numerical solution of an appropriate set of difference equations is an especially powerful method since high speed large capacity digital computers are now available.

The derivation of the difference equations usually proceeds by writing the differential equation at specific points in the continuum and by replacing continuous operators with an appropriate numerical approximation. This method has been extensively developed and is presented in detail by Salvadori and Baron (ref. 7), for example. Houbolt (ref. 8) points out that the derivation of the difference equations directly from the equilibrium equations is straight forward in simple situations; but, for many practical problems the process becomes tedious and may involve uncertain steps. Houbolt proposes that the appropriate difference equations be derived directly, by minimizing the energy expression. Briefly the process advocated by Houbolt consists of dividing the continuum into a regular gridwork, writing the total potential energy of the system in terms of displacements at the grid stations, and minimizing this energy expression with respect to each nodal displacement. Houbolt presents, as an example, the set of difference equations associated with the deflection of a flat rectangular plate.

Walton (ref. 9) utilized the set of difference equations derived directly from the energy expression to determine the bending deflections of flat plates. Schaeffer and Heard (ref. 10) utilized a stress function and the set of difference equations derived from an appropriate energy expression to determine the midplane thermal stress distribution in a flat plate.

The problems considered in references 8, 9, and 10 were linear so that the set of unknown quantitites could be determined by setting the components of gradient expressions to zero and using matrix methods to solve the resulting equations. For nonlinear problems the gradient expressions are nonlinear in terms of the unknowns so that the set of equations which results from setting the gradient components cannot be solved in a simple manner using linear matrix methods. Bogner and others (ref. 11) have utilized direct search methods using a finite element representation of flat plate and mathematical programing techniques with success. The problem of determining the nonlinear displacements of a square plate was considered and the results were found to be in excellent agreement with the results presented by Timoshenko (ref. 2).

The direct minimization of the potential energy expressed in terms of displacements at discrete grid station has not been discussed in the literature.

Symmetric Nonlinear Deflections of Shallow Shells of Revolution

The theory for the bending of thin walled spherical shells has a long history. The fundamental equations were presented by H. Riessner (ref. 12) in 1912 who showed that for non-shallow shells the bending behavior is restricted to a thin region in the immediate vicinity of the boundary layer.

In the shallow shell, on the other hand, the bending effects are, in general, no longer limited to a boundary layer, and asymptotic methods which have been used to solve the deep shell equations are not applicable. Equilibrium equations are developed for a shallow spherical shell in references 13, 14, and 15 on the explicit assumption that the ratio $(H/a)^2$ is small compared to $\frac{H}{a}$ where H is the shell height and a is the radius. A few solutions of the linearized equations are presented by E. Reissner (ref. 14).

The investigation presented by Kaplan (ref. 15) determines the range of nondimensional center deflection $\frac{w_o}{t}$ for a shallow spherical shell, where t is the shell thickness, for which the linear solutions are valid. The range of $\frac{w_o}{t}$ in which the linear solution is valid was found to be very small. For example, at $\mu = 4$, (μ is proportional to $\frac{H/h}{aH}$ where H is the shell height and h is the shell thickness), the equilibrium pressure given by the linear solution is, respectively, 9, 23, and 50 percent too high when $\frac{w_o}{t}$ is 0.1, 0.25, and 0.5. Excellent data from carefully controlled experiments on the large deflections of symmetrically loaded clamped spherical shells for a rather restricted range of the spherical shell parameter, μ, is presented and compared to the solution of the nonlinear equations.

The problem of the clamped shallow spherical shell is the subject of the theoretical investigations presented in references 16 through 20. These solutions which are based on the same nonlinear differential equations are generally in disagreement with each other and with available experimental information. Budiansky (ref. 21) points out that some of the trouble in the solutions presented in references 16 through 20 is due to the wavelike deflection distribution of the shell, which tends to increase with increasing values of μ. Budiansky shows that for values of $\mu > 6$, where the theoretical axisymmetric buckling loads are in very poor agreement with experiment, the lack of correlation is due to initial shell imperfections which precipitate asymmetric buckling modes.

Recently several investigators including Thurston (ref. 22), Archer (ref. 23), and Mescall (ref. 24), have presented results based on the numerical solution of the nonlinear equilibrium equations. The basis for all of these numerical solutions is the replacement of continuous functions with a set of discrete values, and the use of finite-difference approximations of the differential operators. The problem of the clamped spherical shallow shell loaded with a uniform normal pressure is solved by Thurston (ref. 22), using a Newton-Raphson procedure and the results are compared to the experimental results of Kaplan (ref. 15). An iterative method which utilizes a form of Gaussian elimination is presented by Archer (ref. 23) and is applied to the problems of the clamped uniformly loaded spherical cap and the unrestrained shell under point load. A combination of the Gaussian elimination technique and the Newton-Raphson procedure is presented by Mescall (ref. 24) and the resulting method is used to obtain solutions to the uniformly loaded clamped cap and the unrestrained cap with a concentrated load.

An experimental investigation of the effect of boundary conditions on shell buckling is presented by Wang (ref. 25). The investigation considered uniformly loaded spherical caps with both hinged and clamped boundaries and it was concluded that the buckling behavior of shallow caps is influenced by the boundary conditions.

Methods of Unconstrained Minimization

An excellent review of the minimization techniques for nonlinear functions is presented by Spang (ref. 26), where both gradient and random methods of search are discussed. The gradient method appears to be applicable to deterministic problems and is the only general search technique considered in the present investigation. The method of steepest descent which utilizes the negative of the gradient direction as the direction to proceed in minimizing a function was first proposed by Cauchy (ref. 27). In practical applications this method is found to have poor convergence properties. In order to eliminate this problem, methods which employ conjugate gradients have been presented in references 23 through 34.

Rosenbrock in reference 28 presents a modification of steep descent which is reported to have convergence characteristics which are superior to the steep descent method. The algorithm presented by Hestenes (ref. 29) and Beckman (ref. 30) is applicable to linear systems only. Fletcher and Reeves (ref. 3) generalize the algorithm presented by Hestenes for nonlinear problems.

The algorithm presented by Hestenes is described from a geometric point of view by Tomkins (ref. 35) who shows that the algorithm is essentially a sequence of constrained steepest descents. The algorithm presented by Powell (ref. 32) is essentially the same as the algorithm presented by Fletcher and Reeves.

The algorithm which is presented by Davidon (ref. 33) and which is presented in slightly modified form by Fletcher and Powell (ref. 34) employs the concept of a variable metric. In principle this algorithm is similar to the Newton-Raphson technique and is reported to have similar convergence characteristics.

VII. MINIMIZATION OF UNCONSTRAINED FUNCTIONS

The solution of many problems in the field of continuum mechanics can be reduced to the minimization of a function of many variables which has the following form.

$$f(x) = b_i x_i + a_{ij} x_i x_j + c_{ijk} x_i x_j x_k + d_{ijkl} x_i x_j x_k x_l \qquad (3)$$

where x_i are unknowns, b_i, a_{ij}, c_{ijk} and d_{ijkl} are constants. In writing equation (3) the well-known range and summation conventions of index notation have been used (see ref. 36, for example). Functions such as equation (3) which involve products of the unknowns through the fourth degree arise, for example, in studying the behavior of plates and shells when the effects of geometric nonlinearities are included.

It is assumed that the function $f(x)$ and the gradient vector $\frac{\partial f}{\partial x_i}$ can be calculated at any given point x_i. It is also assumed that $f(x)$ is continuous and has continuous derivatives through the second order. Thus, in the neighborhood of the required minimum, h_i, the function $f(x)$ may be expanded by using a Taylor series to give the following:

$$f(x) \cong f(h) + \bar{a}_{ij} \left(h_i - x_i\right) \left(h_j - x_j\right) \qquad (4)$$

where terms involving products of the unknowns of higher than third degree have been neglected. In what follows the function

$f(x)$ will be set equal to the truncated Taylor series for convenience. In writing equation (4) the fact that the gradient vanishes at the minimum has been used, and the matrix of coefficients, $\bar{a}_{ij}$, has been taken to be symmetric. Also note that $\bar{a}_{ij}$ is composed of the matrix $\bar{a}_{ij}$ in addition to appropriate elements of c_{ijk} and d_{ijkl} which are associated with second degree terms in the Taylor series expansion.

Since the second-order terms of the Taylor series expansion dominate in the region of the minimum, methods which exhibit quadratic convergence will converge quickly for the general function. Quadratic convergence means that for a quadratic function the minimum is located exactly, neglecting round-off errors, in a finite number of iterations. The number of iterations is usually equal to or less than the number of unknowns.

It is appropriate to discuss sequential search methods of minimization rather than other methods, such as those utilizing Monte Carlo techniques, for instance, because sequential search methods are quadratically convergent. A sequential search method sets up an algorithm for finding a new approximation of the minimum in terms of the present approximation in the following form:

$$x_i^{\alpha+1} = x_i^{\alpha} + a^{\alpha} p_i^{\alpha} \qquad (5)$$

where a Greek superscript is used to denote the iteration number and does not obey the rules of index notation, and where

a^α is a real number

p_i^α is a vector

The vector p_i^α represents the best direction to proceed from x_i^α where x_i^α is considered as representing the coordinates of a point in an n-dimensional euclidian space where n is the number of unknowns, and the number a^α represents an optimum distance to be traversed from x_i^α in the direction p_i^α. Equation (5) is to be used to generate a sequence of points $x_i^1, x_i^2, \ldots,$ $x_i^\alpha, x_i^{\alpha+1}$ subject to the condition that

$$f\left(x^{\alpha+1}\right) < f\left(x^\alpha\right) \tag{6}$$

There are two basic determinations which must be made in moving from one point to another in the sequence. First, the best direction, p_i^α, must be found; and, second, the optimum distance a^α, for the move along p_i^α from x_i^α must be determined.

Steepest Descent

It is reported that Cauchy (ref. 27) was one of the first to investigate the question of the direction in which to proceed from a given point x_i^α. He reasoned that the direction should be chosen such that the function is decreased most rapidly at x_i^α when proceeding to $x_i^{\alpha+1}$. The direction in which the function changes most rapidly at the point x_i^α can be determined as follows. Consider the points x_i^α and $x_i^\alpha+dx_i$ which lie on the surfaces $f\left(x_i^\alpha\right)$ and $f\left(x_i^\alpha+dx_i\right)$, respectively. The distance between the points x_i^α and $x_i^\alpha+dx_i$ is given by the following relations:

$$ds^2 = dx_i dx_i \tag{7}$$

The rate of change of the function when proceeding along the line between x_i^α and $x_i^\alpha+dx_i$ is given by the following:

$$\frac{df}{ds} = \frac{\partial f}{\partial x_i}\, dx_i \tag{8}$$

It is clear that the change in the function $\frac{df}{ds}$ is completely determined at the point x_i^α by specifying dx_i since the quantities $\frac{\partial f}{\partial x_i}$ are known. Also, the rate of change of the function depends only on the direction of descent at x_i^α and not upon the curvature.

Now let u_i represent the differentials dx_i at x_i^α. It is required to find the value of u_i such that the quantity $\frac{df}{ds}$ becomes an extremum. This problem can be solved by equating all the derivatives of $\frac{df}{ds}$ with respect to u_i to zero, thus;

$$\partial \frac{\partial}{u_i}\left(\frac{df}{ds}\right) = 0 \tag{9}$$

The following equation results after performing the differentation indicated in equation (9) and noting in addition that $\frac{\partial f}{\partial x_i}$ is independent of u_i and that ds is a function of u_i.

$$\frac{\partial f}{\partial x_i} dx_k dx_k = \frac{\partial f}{\partial x_j} dx_j dx_i \tag{10}$$

Equation (10) can be represented by the following form

$$dx_i = \text{Const.} \frac{\partial f}{\partial x_i} \tag{11}$$

since i is the only free index in equation (10). Equation (11) may thus be interpreted as saying that a necessary condition for maximum ascent or descent from the point x_i^α on the surface $f\left(x_i^\alpha\right)$ is that the differentials dx_1 be proportional to the known partial derivatives $\frac{\partial f}{\partial x_i}$ evaluated at the point x_i^α. Since the quantities $\frac{\partial f}{\partial x_i}$ represent the gradient of the function at x_i^α it follows that a move from x_i^α in the direction of the gradient of the function evaluated at x_i^α will lead to the greatest change in the function in the immediate neighborhood of the point x_i^α. Thus if the direction p_i^α is chosen as the negative gradient.

$$p_i^\alpha = - \frac{\partial f}{\partial x_i}\left(x^\alpha\right) \tag{12}$$

the function will undergo the most rapid decrease in the vicinity of x_i^α.

The sequential search technique which utilizes the negative gradient as the direction p_i^α, and which subsequently determines a^α by locating the minimum of the function along the line

$x_i^\alpha + a^\alpha p_i^\alpha$ is termed the method of steepest descent. The sequential search technique is termed the method of optimum gradients if the negative gradient is used as the direction p_i^α and a^α is determined so that the inequality (6) is satisfied. The numerical procedure termed steepest descent has been widely used especially in the field of optimum control (see refs. 37 and 38 for instance). However, the method of steepest descent as used in the above references and as defined above does not converge quadratically. As a matter of fact, steepest descent methods are known to converge very slowly near the minimum of the function. In addition, this method is very inefficient in minimizing a function which, when represented as a hypersurface in euclidian n-space, has the character of a long narrow curved or winding valley, because short zig-zag steps across the valley rather than long steps down the valley are taken.

The Method of Conjugate Directions

It is shown by Tomkins (ref. 35) that the steepest descent algorithm, as outlined above, can be modified so as to be quadratically convergent. The steepest descent method is modified by requiring that each direction p_i^α of the sequence be conjugate to the direction $p_i^{\alpha-1}$ of the previous move where the term conjugate is taken to mean that the vectors satisfy a general orthogonality condition. The terminology a-conjugate will also be used in the following development and indicates that the vectors satisfy an orthogonality condition with respect to the metric a_{ij}. The method of conjugate directions was originally presented by Hestenes (ref. 29) as a method of solving the equivalent problems of finding the solutions to a set of simultaneous equations or of finding the location of the minimum of a suitably defined quadratic function. It was subsequently shown by Fletcher and Reeves (ref. 31) that the general algorithm presented by Hestenes when suitably modified is an efficient algorithm for locating the unconstrained minimum of an object function having the form of equation (3). For the example presented by Fletcher and Reeves (ref. 31) the method of conjugate gradients is shown to be far superior to the method of steepest descent. The algorithm for the method of conjugate directions, as presented by Hestenes is as follows.

Consider a set of linear algebraic equations which is written in the following form using index notation.

$$a_{ij}x_j = - b_i \tag{13}$$

It is assumed that a solution h_i of the system of equation (13) exists.

Assume that a set of n a-conjugate vectors p_i^{α} $(\alpha = 1, 2, . . ., n)$, $(i = 1, 2, . . ., n)$ exists, thus $p_i^{\alpha} p_j^{\beta} a_{ij} = 0$ for $\alpha \neq \beta$. Furthermore, a_{ij} will be restricted to be positive definite so that for $\alpha = \beta$ the product

$p_i^{\alpha} p_j^{\alpha} a_{ij} > 0$. Since the elements p_i^{α} are a-conjugate, the vectors are linearly independent and form a basis in n-dimensional euclidian space. The set of a-conjugate vectors are independent since if $p_i^{\beta} = \lambda p_i^{\alpha}$ then $\lambda p_i^{\alpha} p_j^{\alpha} a_{ij} \neq 0$ and does not satisfy the orthogonality condition. The solution vector h_i can thus be written as a linear combination of the p_i^{α} as follows:

$$h_i = c^1 p_i^{\,1} + c^2 p_i^{\,2} + \ldots + c^n p_c^{\,n} \tag{14}$$

Clearly, the solution h_i is determined if the c^{α} are known. The c^{α} follow directly by multiplying each side of equation (14) by $a_{ij} p_j^{\alpha}$, and by noting that the directions p_j^{α} are a-conjugate. Thus,

$$a_{ij} h_j p_i^{\alpha} = c^{\alpha} p_i^{\alpha} p_j^{\alpha} a_{ij} \tag{15}$$

and consequently

$$c^{\alpha} = -b_i p_i^{\alpha} \big/ a_{ij} p_i^{\alpha} p_j^{\alpha} \tag{16}$$

where use has been made of equation (13). The solution can now be written as follows:

$$h_l = -\left\{ \frac{b_i p_i^{\,1}}{a_{ij} p_i^{\,1} p_j^{\,1}} p_l^{\,1} + \ldots + \frac{b_i p_i^{\,n}}{a_{ij} p_i^{\,n} p_j^{\,n}} p_l^{\,n} \right\} \tag{17}$$

The method outlined above has been developed by using a general set of a-conjugate directions. A very elegant algorithm results if the set of directions are chosen so as to be related to the gradients. This method which is called the method of conjugate gradients is described below.

The Method of Conjugate Gradients

Equation (13) is an equality only when the vector x_i corresponds to the solution vector h_i. In general, for $x_i \neq h_i$, it is proper to define the residual vector as follows:

$$r_i^{\alpha} = -\left(a_{ij}x_j^{\alpha} + b_i\right) \tag{18}$$

where r_i is termed the residual vector. When $x_i = h_i$ the residual vector is a null vector.

The method of conjugate gradients, which was evolved by Hestenes and explained by Beckman,is derived by forming two conjugate sequences by the Gram-Schmidt orthogonalization procedure: a set of residual vectors, r_i^{α}, which are orthogonal in the ordinary sense that $r_j^{\alpha}r_j^{\alpha} = 0$ for $\alpha \neq \beta$ and is not equal to zero for $\alpha = \beta$, and a set p_i^{α} which is a-conjugate. The residual vectors and the direction vectors are formed in the order: $r_i^1, p_i^1; r_i^2, p_i^2; \ldots; r_i^n, p_i^n$.

It is assumed that one of the sets, r_i^{α} for instance, has been formed. The elements r_i^{α} are by definition orthogonal and are thus linearly independent. Using the set of vectors r_i^{α} a set of a-conjugate vectors p_i^{α} can be determined as follows:

Let $p_i^1 = r_i^1$: p_i^2 can now be chosen as a linear combination of r_i^2 and p_i^1, $p_i^2 = r_i^2 + c^{21}p_i^1$. The vector p_i^2 will be a-conjugate to p_i^1 if the constant c^{21} is chosen so that the vector $a_{ij}p_j^2$ is orthogonal to the vector p_i^1, thus

$$p_i^2 p_j^1 a_{ij} = 0 = a_{ij}r_j^2 p_i^1 + c^{21} p_i^1 p_j^1 a_{ij} \tag{19}$$

and consequently

$$c^{21} = -a_{ij}r_j^2 p_i^1 \Big/ a_{ij}p_i^1 p_j^1 \tag{20}$$

In general, after having determined p_i^1, p_i^2, . . ., p_i^{α} ($\alpha<n$), the vector $p_i^{\alpha+1}$ can be written in the following form.

$$p_i^{\alpha+1} = r_i^{\alpha+1} + c^{\alpha+1,1}p_i^1 + \ldots + c^{\alpha+1,\alpha}p_i^{\alpha} \tag{21}$$

Equation (21) is multiplied by $a_{ij}p_j^{\alpha}$ to give

$$0 = a_{ij}p_j^{\alpha} r_i^{\alpha+1} + c^{\alpha+1,\alpha}a_{ij}p_j^{\alpha} p_i^{\alpha} \tag{22}$$

where the condition of orthogonality has been used. It follows that the coefficient $c^{\alpha+1,\alpha}$ is given by the following

$$c^{\alpha+1,\alpha} = -a_{ij}p_j^{\alpha} r_i^{\alpha+1} \Big/ a_{ij}p_j^{\alpha} p_i^{\alpha} \tag{23}$$

A set of a-conjugate directions p_i^{α} can therefore be developed from the orthogonal set r_i^{α} by application of the following recursion formula:

$$p_l^{\alpha+1} = r_l^{\alpha+1} - \left\{ \frac{a_{ij} r_j^{\alpha+1} p_i^{1}}{a_{ij} p_i^{1} p_j^{1}} p_l^{1} + \ . \ . \ . + \frac{a_{ij} r_j^{\alpha+1} p_i^{\alpha}}{a_{ij} p_i^{\alpha} p_j^{\alpha}} p_l^{\alpha} \right\} \qquad (24)$$

A recursion relation for the set of orthogonal residual vectors r_i^{α} can be derived in a similar manner by assuming that a set of a-conjugate vectors p_i^{α} are available. The $\alpha+1$ element in the set of residual vectors is then taken in the following form.

$$r_i^{\alpha+1} = r_i^{\alpha} + t^{\alpha+1,1} a_{ij} p_j^{1} + \ . \ . \ . + t^{\alpha+1,\alpha} a_{ij} p_j^{\alpha} \qquad (25)$$

Equation (20) is multiplied by r_i^{α} to give.

$$0 = r_i^{\alpha} r_i^{\alpha} + t^{\alpha+1,1} a_{ij} p_j^{1} r_i^{\alpha} + \ . \ . \ . + t^{\alpha+1,\alpha} a_{ij} p_j^{\alpha} r_i^{\alpha} \qquad (26)$$

where the condition of orthogonality has been used. It follows that the coefficient $t^{\alpha+1,\alpha}$ is given by the following

$$t^{\alpha+1,\alpha} = -r_i^{\alpha} r_i^{\alpha} \big/ a_{ij} p_j^{\alpha} r_i^{\alpha} \qquad (27)$$

It will now be shown that $a_{ij}p_i^{\alpha}r_j^{\beta} = 0$ for $\beta \neq \alpha+1$ or $\beta \neq \alpha$. The error $r_i^{\alpha+1}$ from equation (18) is as follows:

$$r_i^{\alpha+1} = -\left(a_{ij}x_j^{\alpha+1} + b_i\right) \tag{28}$$

The substitution of equation (5) into equation (28) gives the following:

$$r_i^{\alpha+1} = r_i^{\alpha} - a^{\alpha}a_{ij}p_j^{\alpha} \tag{29}$$

The multiplication of each term of equation (29) by r_i^{β} gives the following:

$$r_i^{\alpha+1}r_i^{\beta} = r_i^{\alpha}r_i^{\beta} - a^{\alpha}a_{ij}p_j^{\alpha}r_i^{\beta} \tag{30}$$

Since the vector r_i^{α} is an element of an orthogonal set it follows directly from equation (30) that the quantity $a_{ij}p_j^{\alpha}r_i^{\beta} = 0$ for $\beta = \alpha$ or $\beta = \alpha+1$. Thus equations (24) and (25) can be written in the following form:

$$p_i^{\alpha+1} = r_i^{\alpha+1} - \beta^{\alpha}p_i^{\alpha} \tag{31}$$

and

$$r_i^{\alpha+1} = r_i^{\alpha} - a^{\alpha} a_{ij} p_j^{\alpha} \tag{32}$$

where

$$\beta^{\alpha} = a_{ij} r_i^{\alpha+1} p_j^{\alpha} \Big/ a_{ij} p_j^{\alpha} p_i^{\alpha} \tag{33}$$

$$a^{\alpha} = r_i^{\alpha} r_i^{\alpha} \Big/ a_{ij} p_j^{\alpha} r_i^{\alpha} \tag{34}$$

Notice that r_i^2 can be determined from equation (32) since r_i^1 and p_i^1 are known $\left(\text{recall that } p_i^1 = r_i^1\right)$. The remaining elements of each of the conjugate sets r_i^{α}, $a_{ij}\ p_j^{\alpha}$ can now be determined by recursively using equations (31) and (32). The coefficient a^{α} in equation (5) is determined by substituting equation (5) into equation (18) and comparing the resulting form to equation (32). It is determined that a^{α} is as defined by equation (34), so that the a^{α} used in equations (5) and (32) are identical.

The basic algorithm for the conjugate gradient method is summarized as follows. Let x_i^1 be an arbitrary approximation of the solution vector, h_i. The following formulas describe the fundamental iterative procedure which will, neglecting round-off error, lead to the solution of the set of linear equations represented by equation (13).

$$p_i^{1} = r_i^{1} \tag{35}$$

$$x_i^{\alpha+1} = x_i^{\alpha} + a^{\alpha} p_i^{\alpha} \tag{36}$$

$$r_i^{\alpha+1} = r_i^{\alpha} - a^{\alpha} a_{ij} p_j^{\alpha} \tag{37}$$

$$p_i^{\alpha+1} = r_i^{\alpha+1} + c^{\alpha} p_i^{\alpha} \tag{38}$$

where:

$$a^{\alpha} = r_i^{\alpha} r_i^{\alpha} \Big/ a_{ij} p_j^{\alpha} r_i^{\alpha} \tag{39}$$

$$c^{\alpha} = -a_{ij} r_i^{\alpha+1} p_j^{\alpha} \Big/ a_{ij} p_i^{\alpha} p_j^{\alpha} \tag{40}$$

The method of conjugate gradients as summarized above is discussed by Tomkins (ref. 35) from the geometric point of view, as a method of locating the minimum of a hypersurface defined in euclidian n-dimensional space. The hypersurface is defined by the following relation

$$f(x) = a_{ij}\left(h_i - x_i\right)\left(h_j - x_j\right) \tag{41}$$

The gradient of this function, for symmetric a_{ij} is given by

$$g_i^\alpha = 2a_{ij}\left(h_j - x_j^\alpha\right) \tag{42}$$

By noting equations (13) and (18) it is seen that g_i^α is proportional to the residual vector r_i^α.

Sequential Search Methods For Locating the Minimum of Nonquadratic Functions

Two algorithms have been reported in the literature as being effective methods for locating the unconstrained minimum of non-quadratic functions of the form given by equation (3). One of these is the variable metric method which was evolved by Davidon (ref. 33) and which was presented in slightly modified form by Fletcher and Powell (ref. 34). The other is the method of conjugate gradients as presented by Fletcher and Reeves (ref. 31). Each of these algorithms is described below.

Conjugate Gradient Method.- The method of conjugate gradients as presented by Hestenes (ref. 29) and Beckman (ref. 30) is an elegant method for solving the equivalent problems of determining the solution of a set of simultaneous equations or of locating the minimum of a quadratic function. The method is quadratically convergent so that, neglecting round-off errors, the solution will be determined in n steps. The applicability of the basic conjugate gradient algorithm has been evaluated for finding the unconstrained minimum of a nonquadratic function by Fletcher and Reeves (ref. 31). The algorithm presented by Fletcher and Reeves is very similar to that presented Hestenes. The basic difference is that the coefficient a^α in equation (5) is not directly determinable for a nonquadratic function, but must be located using a numerical technique. The value of a^α for the nonquadratic function is taken such that the function $f(x)$ is a minimum along the line $x_i^\alpha + a^\alpha p_i^\alpha$. That is a^α is equal to the value of λ which satisfies the following equality.

$$\frac{\partial f}{\partial \lambda}\left(x^\alpha+\lambda p^\alpha\right) = 0 \tag{43}$$

For an arbitrary initial starting point x_i^1, set

$$p_i^1 = -g_i^1 \tag{44}$$

where g_i^α is the gradient of function represented by equation (3), thus

$$g_i^\alpha = \frac{\partial f}{\partial x_i}\left(x^\alpha\right)$$

The new point $x_i^{\alpha+1}$ is given by

$$x_i^{\alpha+1} = x_i^\alpha + a^\alpha p_i^\alpha \tag{45}$$

where a^α is chosen so that equation (43) is satisfied. The new direction $p_i^{\alpha+1}$ is then determined from the following relation

$$p_i^{\alpha+1} = -g_i^{\alpha+1} + c^\alpha p_i^\alpha \tag{46}$$

where

$$c^\alpha = g_i^{\alpha+1} g_i^{\alpha+1} \Big/ g_j^\alpha g_j^\alpha \tag{47}$$

It can be shown that the magnitude of c^α as determined from equation (47) is the same as the magnitude of c^α as determined from equation (40) for a quadratic function.

Fletcher and Reeves leave one with the impression that the algorithm which is represented by equations (36) through (39) is valid only if the matrix $\bar{a}_{ij}$ has many zero elements. In fact, however, validity and applicability of the algorithm is independent of the structure of the matrix $\bar{a}_{ij}$, except that $\bar{a}_{ij}$ must be symmetric.

Variable Metric Method of Minimization.- The variable metric method of locating the unconstrained minimum of a function of several variables was evolved by Davidon (ref. 33) and was further refined by Fletcher and Powell (ref. 34).

Consider a function $f(x)$ where the set x_i specifies the coordinates of a point in an n-dimensional linear space, L_1. The set x_i for which $f(x)$ is constant forms an n-1 dimensional surface in this space. One surface, of the family of surfaces, passes through each point x_i, and the surface in the neighborhood of the points is characterized by the gradient $g_i = \frac{\partial f(x)}{\partial x_i}$. The components of the gradient can be interpreted as defining a point in another linear space L_2.

Consider now a function which is continuous and which as continuous derivatives through the second order. In the neighborhood of the point a in L_1 the function $f(x)$ can be represented by a Taylor series expansion about the point a as follows

$$f(x) = f\left(x^a\right) + \frac{\partial f\left(x^a\right)}{\partial x_i}\left(x_i - x_i{}^a\right) + \frac{1}{2}\frac{\partial^2 f\left(x^a\right)}{\partial x_i \partial x_j}\left(x_i - x_i{}^a\right)\left(x_j - x_j{}^a\right) + \text{H.O.T.} \tag{48}$$

The gradient g_i is as follows

$$g_i(x) \cong \frac{\partial f\left(x^a\right)}{\partial x_i} + \frac{\partial^2 f\left(x^a\right)}{\partial x_i \partial x_j}\left(x_j - x_j{}^a\right) \tag{49}$$

where terms of second degree and higher in x have been neglected. For convenience, the expression given by equation (49) will be treated as an equality in what follows.

It follows directly from equation (49) that the change of the gradient dg_i is given by the following expression

$$dg_i(x) = \frac{\partial^2 f\left(x^a\right)}{\partial x_i \partial x_j}\, dx_j \tag{50}$$

Equation (50) specifies a linear mapping of changes of position dx_i in L_1 onto a corresponding change in gradient dg_i in L_2. If the matrix $\frac{\partial^2 f}{\partial x_i \partial x_j}$ were independent of x_i then the value of the gradient at one point would be sufficient to determine the minimum. In that case the desired change in the gradient is $-g_i$ so that the associated change in x_i could be found by inverting the matrix $\frac{\partial^2 f}{\partial x_i \partial x_j}$. However, for an object function of greater then second degree in the unknowns the matrix $\frac{\partial^2 f}{\partial x_i \partial x_j}$ is not constant. Also, explicit evaluation of the matrix and its subsequent inversion (which is in essence the Newton-Raphson procedure which will be discussed later) at a point which is far from the minimum of the function may not represent an optimum expenditure of computational effort.

Instead of explicitly evaluating the inverse of $\frac{\partial^2 f}{\partial x_i \partial x_j}$, the inverse will be assumed. The approximation of the inverse which will be designated as H_{ij}^{α} specifies a linear mapping of changes in the gradient in L_2 onto corresponding changes in position L_1. This matrix which is the metric of the L_2 space is restricted to be symmetric and positive definite.

An algorithm for systematically improving the metric so that the minimum of the function is determined in a finite number of steps is now required. It is specified that the improved metric designated $H_{ij}^{\alpha+1}$ is to be formed such that all the information contained in H_{ij}^{α} is retained and in addition has the following property.

$$H_{ij}^{\alpha+1} \Delta g_j^{\alpha} = \Delta x_i^{\alpha} \tag{51}$$

where

$$\Delta x_i^{\alpha} = x_i^{\alpha+1} - x_i^{\alpha} = a^{\alpha} p_i^{\alpha} \tag{52}$$

and

$$\Delta g_i^{\alpha} = g_i^{\alpha+1} - g_i^{\alpha} \tag{53}$$

The coefficient a^{α} is chosen such that the object function is a minimum along the line defined by equation (52). The substitution of equation (52) into (51) gives the following

$$H_{ij}^{\alpha+1} \Delta g_i^{\alpha} = a^{\alpha} p_i^{\alpha} \tag{54}$$

It is now required to form the matrix $H_{ij}^{\alpha+1}$ in such a way that equation (54) is satisfied and so that the information contained in H_{ij}^{α} is retained. The simplest way of satisfying the last requirement is to consider $H_{ij}^{\alpha+1}$ as the matrix H_{ij}^{α} plus a correction factor, or factors, such that equation (54) is satisfied. Consider therefore $H_{ij}^{\alpha+1}$ in the following form

$$H_{ij}^{\alpha+1} = H_{ij}^{\alpha} + A_{ij}^{\alpha} + B_{ij}^{\alpha} \tag{55}$$

The substitution of equation (55) into equation (54) gives the following

$$\left(H_{ij}^{\alpha} + A_{ij}^{\alpha} + B_{ij}^{\alpha}\right) \Delta g_j^{\alpha} = a^{\alpha} p_i^{\alpha} \tag{56}$$

In equation (56) all the quantities are known except the matrices A_{ij}^{α} and B_{ij}^{α}. These matrices are determined as follows.

The vector

$$t_i^{\alpha} = H_{ij}^{\alpha} \Delta g_j^{\alpha} \tag{57}$$

is not in general colinear with p_i^{α}. Therefore, choose B_{ij}^{α} such that

$$B_{ij}^{\alpha} \Delta g_j^{\alpha} = -t_i^{\alpha} \tag{58}$$

and A_{ij}^{α} such that

$$A_{ij}^{\alpha} \Delta g_j^{\alpha} = a^{\alpha} p_i^{\alpha} \tag{59}$$

It can be shown by direct substitution into the appropriate equation (58) or (59) that acceptable forms for A_{ij}^{α} and B_{ij}^{α} are as follows

$$A_{ij}^{\alpha} = \frac{a^{\alpha} p_i^{\alpha} p_j^{\alpha}}{p_k^{\alpha} \Delta g_k^{\alpha}} \tag{60}$$

$$B_{ij}^{\alpha} = - \frac{t_i^{\alpha} t_j^{\alpha}}{\Delta g_k^{\alpha} t_k^{\alpha}} \tag{61}$$

where t_i^{α} is given by equation (57):

The expression for the improved matrix, equation (55) thus becomes

$$H_{ij}^{\alpha+1} = H_{ij}^{\alpha} + \frac{a^{\alpha}}{p_k^{\alpha} \Delta g_k^{\alpha}} p_i^{\alpha} p_j^{\alpha} - \frac{H_{ik}^{\alpha} H_{jl}^{\alpha} \Delta g_k^{\alpha} \Delta g_l^{\alpha}}{H_{mn}^{\alpha} \Delta g_m^{\alpha} \Delta g_n^{\alpha}} \tag{62}$$

The new direction $p_i^{\alpha+1}$ is taken to be

$$p_i^{\alpha+1} = -H_{ij}^{\alpha+1} g_i^{\alpha+1} \tag{63}$$

since from equation (50) and (52)

$$p_i^{\alpha+1} = \left[\frac{\partial^2 f}{\partial x_i \partial x_j}^{\alpha+1} \right]^{-1} \Delta g_j^{\alpha+1} \tag{64}$$

and $H_{ij}^{\alpha+1}$ has been identified as the approximation to the inverse of the matrix $\left[\frac{\partial^2 f}{\partial x_i \partial x_j} \right]$.

For the general case where the object function is nonquadratic, the coefficient a^{α} must be found by a numerical technique whereby the minimum along the line $x_i^{\alpha} + \lambda p_i^{\alpha}$ is located. For the case of a quadratic function however, a^{α} is determined explicitly from the following equation

$$a^{\alpha} = - \frac{g_i^{\alpha} p_i^{\alpha}}{a_{ij} p_i^{\alpha} p_j^{\alpha}} \tag{65}$$

where

$$a_{ij} = \frac{\partial^2 f\left(x^{\alpha}\right)}{\partial x_i \partial x_j} \tag{66}$$

It is shown by Fletcher and Powell that the algorithm presented above finds the minimum of a quadratic function in n-iterations. It is also shown that after n-iterations the matrix H_{ij}^{n} is the inverse of the matrix a_{ij} given by equation (66).

The generalized Newton-Raphson method.- The generalized Newton-Raphson method is a sequential method of locating the minimum of a function or of solving a set of nonlinear equations. However, the term sequential does not ordinarily connote the same meaning as was implied in the discussion of the methods of conjugate gradients.

Consider once more a general function $f(x)$. It was shown in the previous section that if $f(x)$ and its derivatives through the second order are continuous at the point x_i^a the Taylor series expansion is given by equation (48) and the gradient by equation (49), which is repeated below

$$g_i(x) = \frac{\partial f\left(x^a\right)}{\partial x_i} + \frac{\partial^2 f\left(x^a\right)}{\partial x_i \partial x_j} \left(x_j - x_j^a\right) + \text{H.O.T.} \tag{49}$$

Let us now assume that the solution is located at $\bar{h}_i$. Thus since a necessary condition for a relative minimum is that the gradient vanish it follows that

$$\frac{\partial f\left(x^a\right)}{\partial x_i} + \frac{\partial^2 f\left(x^a\right)}{\partial x_i \partial x_j} e_j^{\ a} = 0 \tag{67}$$

where terms of greater then the first degree in x have been dropped and where

$$e_j^{\ a} = \left(\bar{h}_j^{\ a} - x_j^{\ a}\right) \tag{68}$$

Since the quantities $\frac{\partial f}{\partial x_i}$ and $\frac{\partial^2 f}{\partial x_i \partial x_j}$ are known at $x_j^{\ a}$, the error vector $e_j^{\ a}$ can be found by linear matrix methods as follows

$$e_j^{\ a} = - \left[\frac{\partial^2 f\left(x^a\right)}{\partial x_i \partial x_j}\right]^{-1} \frac{\partial f\left(x^a\right)}{\partial x_i} \tag{69}$$

where $\left[\frac{\partial^2 f}{\partial x_i \partial x_j}\right]^{-1}$ is the inverse of the $\left[\frac{\partial^2 f}{\partial x_i \partial x_j}\right]$.

If the error $\left|e_j^{\ a}\right|$ is larger than a preassigned small number, δ, then a new initial point is taken as:

$$x_i^{\ a+1} = x_i^{\ a} + e_i^{\ a} \tag{70}$$

Equation (69) is then reapplied until the convergence criterion is satisfied. The method is presented by McGill (ref. 39), together with general conditions which, if satisfied, guarantee that the sequence of solutions will converge.

The algorithm as presented McGill (ref. 39) and others in the literature has a serious drawback. The satisfaction of equation (69) means only that the solution vector is associated with a stationary point on the hypersurface $f(x)$. The point may be a saddle point or a maximum rather than a minimum. In order to guarantee that the solution is associated with a minimum it is necessary that the following condition be satisfied for each sequence of points

$$f\left(x^{a+1}\right) < f\left(x^{a}\right) \tag{71}$$

The One-Dimensional Search

In each of the techniques discussed in the foregoing section it is tacitly assummed that the parameter a^{α} which appears in equations (5), (45), and (52) can be found in an efficient manner such that the orthogonality condition

$$g_i^{\alpha+1} p_i^{\alpha} = \left|g_i^{\alpha+1}\right| \cdot \left|p_i^{\alpha}\right| \cos \varphi = 0 \tag{72}$$

is satisfied, where φ is the angle between the gradient vector, and the conjugate gradient vector, and the notation $\left|\quad\right|$ indicates the absolute value. Various methods, which have been discussed elsewhere in the literature, are presented below.

In general, the relation (72) cannot be exactly satisfied because of round-off error. Thus a^{α} is actually determined such that $\varphi = \frac{\pi}{2} \pm \epsilon$ where ϵ is a preassigned small positive number. It follows directly from a well-known trigonometric identity that

$$\cos\left(\frac{\pi}{2} \pm \epsilon\right) = \pm \sin \epsilon \tag{73}$$

and further, for $\epsilon \ll 1$ it follows that

$$\cos\left(\frac{\pi}{2} \pm \epsilon\right) \simeq \pm \epsilon \tag{74}$$

Thus if a^{α} is chosen such that

$$\frac{\left| g_i^{\alpha+1} p_i^{\alpha} \right|}{\left| g_j^{\alpha+1} \right| \cdot \left| p_j^{\alpha} \right|} \leq \epsilon \tag{75}$$

the angle φ will differ from the desired $\frac{\pi}{2}$ radians by not more than ϵ radians. Theoretically ϵ could be made as small as desired, but practically for a given digital computer ϵ will have a lower limit because of round-off error.

The overall rate at which the conjugate gradient method converges to a minimum of the object function is, to some degree, related to the efficiency of the one-dimensional search technique.

The one-dimensional search should be capable of finding the a^{α} subject to the condition given by equation (75) with as few function and gradient calculations as possible.

A typical plot of the object function with a parameter λ is shown in figure 1, where

$$f(x) = f\left(x^{\alpha}+\lambda p^{\alpha}\right) \tag{76}$$

where $f(x)$ is the variation of the function from point x_i^{α} in the direction of p_i^{α} and where x^{α}, $f\left(x^{\alpha}\right)$, p_i^{α} and g_i^{α} are known at the start of a one-dimensional search. The required value λ_m is that value of λ for which the inequality (75) is satisfied.

The particular technique chosen to find λ_m is dependent on the manner in which the problem is specified, that is, whether or not the gradient components are analytically prescribed or are calculated by finite-difference approximations. Whatever the method, however, the calculation will consist of three parts: the estimation of λ_m; the bounding of the minimum; and the interpolation of λ_m until a criterion is satisfied.

Estimation of λ_m.- To a great degree, the efficiency of the one-dimensional search technique is dependent on the initial estimate of λ_m since an initial estimate which is too small will lead to an excessive number of calculations in order to bound the minimum. In order to obtain an estimate of λ_m suppose that an

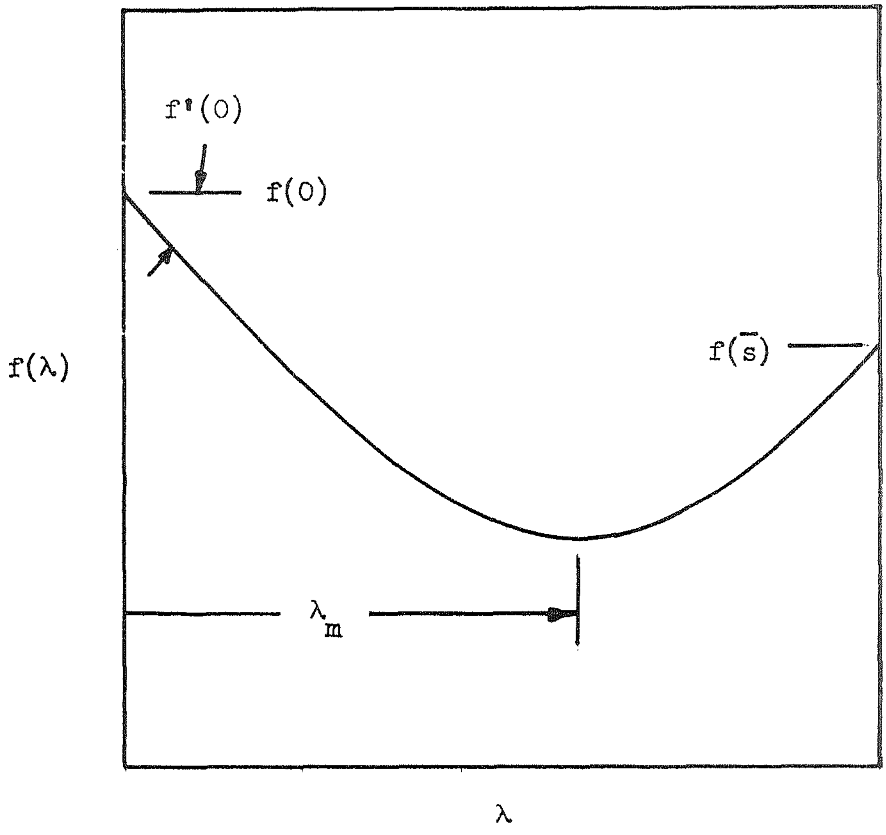

Figure 1.- One dimensional variation of object function.

estimate, est, of the minimum value of $f(x)$ is available. Then, by assuming that the estimate is correct, that the minimum lies along the line $x_i^\alpha+\lambda p_i^\alpha$, and that the function $f(x)$ is quadratic the value λ_m is given by the following equation:

$$\left(\lambda_m\right)_{est} = 2\left(est-f\left(x^\alpha\right)\right)\Big/p_i^\alpha g_i^\alpha \tag{77}$$

Bounding the minimum.- A tentative step length $s = \left(\lambda_m\right)_{est}$ is taken and the slope $f' = \frac{df}{d\lambda}$ is examined at the points $\lambda = s, 2s, 4s, \ . \ . \ .$ until the slope f' is nonnegative. Let $\overline{s}$ be the value of λ associated with the point at which f' becomes nonnegative, then the minimum lies in the interval $0 < \lambda_m < \overline{s}$.

Cubic interpolation.- Since the function values $f(0)$, $f(\overline{s})$ and the slopes $f'(0)$, $f'(\overline{s})$ are readily available it is appropriate to use all of this information to find an interpolated value of the minimum. A cubic interpolation method first proposed by Davidon and subsequently utilized by various investigators utilizes all of these known quantities. In this method a cubic curve is passed through the given points with the prescribed slopes and then the minimum is estimated by the following relationship

$$\left(\lambda_m\right)_{est} = \overline{s}\left\{1-\left[\frac{f'(\overline{s})+y-\tau}{f'(\overline{s})-f'(0)+2y}\right]\right\} \tag{78}$$

where

$$\tau = 3\left[\frac{f(0)-f(\bar{s})}{\bar{s}}\right] + f'(0) + f'(\bar{s}) \tag{79}$$

$$y = \left(\tau^2 - f'(0)\ f'(\bar{s})\right)^{\frac{1}{2}} \tag{80}$$

It is shown by Davidon that the root given by equation (78) lies in the interval $0 \leq \lambda \leq \bar{s}$ . .

By repeatedly applying the above interpolation scheme, the orthogonality constraint given by equation (74) can be satisfied. However, actual calculations have indicated that, because of round-off error, no improvement is made after the second or third interpolation.

Quadratic interpolation.- For the case where the gradient components are calculated numerically using a finite-difference approximation, it becomes time consuming to calculate all the derivatives which are necessary in the cubic interpolation scheme. For this case it is more efficient to first bound the minimum by finding some value of λ such that $f(\lambda)>f(0)$. Then, using a finite difference approximation the slope $f'(0)$ is calculated. The interpolated value of the minimum is then given by the following equation

$$\lambda_m = - \frac{f'(0)}{2\gamma} \tag{81}$$

where

$$\gamma = \frac{1}{\bar{s}}\left\{f(\bar{s})-f(0)-\bar{s}f'(0)\right\} \tag{82}$$

As in the case of cubic interpolation, the orthogonality condition equation (74) can be satisfied by repeated application of equation (81). However, very limited experience with this method indicates that it is more efficient to accept the first value of λ_m given by equation (81), provided that the relation

$$f\left(\lambda_m\right) < f(0) \tag{83}$$

is satisfied.

VIII. THE NONLINEAR BEHAVIOR OF A SHALLOW SHELL OF REVOLUTION SUBJECT TO AXISYMMETRIC LOADS AND ARBITRARY EDGE RESTRAINTS

The object function for a conservative elasticity problem such as a shallow shell is the potential energy of the shell. The set of displacements corresponding to the shell equilibrium state is then that set which makes the potential energy a minimum.

The shell geometry is shown by figure 2. The shape is taken to be a shallow surface of revolution, and is not restricted to be a spherical shallow shell. The thickness, t, and the modulus of elasticity, E, may vary along the shell meridion. The displacements $\bar{u}$, $\bar{w}$, are positive as shown. The displacements $\bar{u}$ and $\bar{w}$, and the slope, $\frac{\partial \bar{w}}{\partial r}$, may be elastically restrained at the shell boundary. The loading is taken to be axisymmetric, however, it is assumed that no concentrated force exists at the center of the shell.

Strain Potential

The strain energy for an isotropic elastic shell is given by the following (see ref. 40, for example).

$$U = \int_{0}^{2\pi}\int_{r_1}^{r_2}\int_{-\frac{h}{2}}^{\frac{h}{2}} \frac{E}{2\left(1-\nu^2\right)} \left\{ \epsilon_r^{\ 2} + \epsilon_\theta^{\ 2} + 2\nu\epsilon_r\epsilon_\theta + \frac{1}{2}\left(1-\nu\right)\gamma_{r\theta}^{\ \ 2} \right\} rdzdrd\theta \tag{84}$$

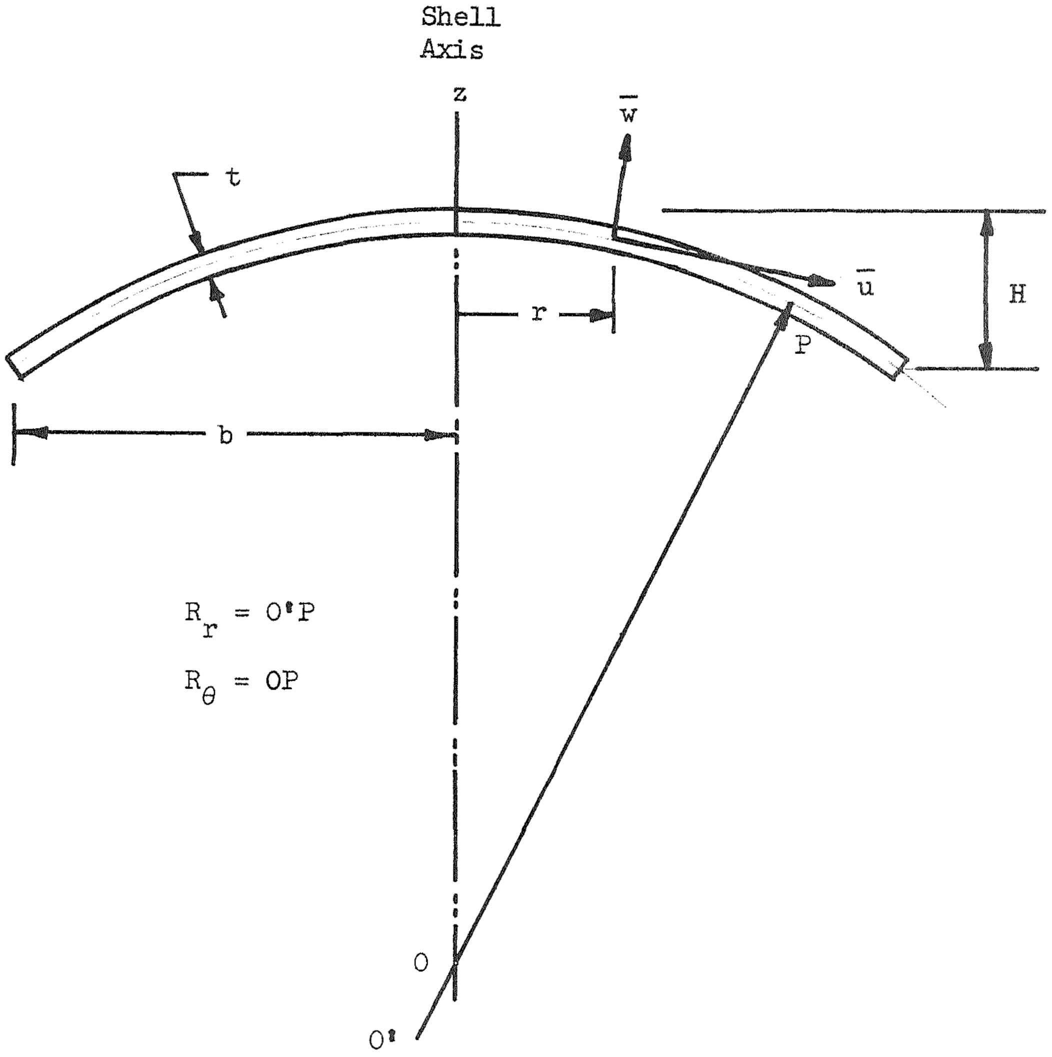

Figure 2.- Shallow shell geometry.

where:

ϵ_r, ϵ_θ are physical normal strains in the r and θ coordinate direction

$\gamma_{r\theta}$ is the physical shear strain

r, θ are orthogonal surface coordinates

z is the coordinate normal to the surface

ν is Poisson's ratio

Equation (84) can be readily integrated with respect to the z coordinate by assuming that the strains can be represented as follows:

$$\left.\begin{aligned} \epsilon_r &= e_r + z\,k_r \\ \epsilon_\theta &= e_\theta + z\,k_\theta \\ \gamma_{r\theta} &= e_{r\theta} + z\,k_{r\theta} \end{aligned}\right\} \quad (85)$$

The quantities $(e_r, e_\theta, e_{r\theta})$ are the middle surface strains while $(\epsilon_r, \epsilon_\theta, \gamma_{r\theta})$ are the strains at any point in the shell. The quantities $(k_r, k_\theta, k_{r\theta})$ are distortions of the middle surface which are closely related to the changes in the shell curvature caused by bending.

The strain energy expression (84) after substitution of equation (85) and integration with respect to z can be represented as the sum of two terms, the membrane energy U_M and the bending energy U_B

$$U = U_M + U_B \quad (86)$$

where:

$$U_M = \frac{1}{2} \int_0^{2\pi} \int_{r_1}^{r_2} K \left\{ e_r^2 + e_\theta^2 + 2\nu e_r e_\theta + \frac{1}{2}(1-\nu) e_{r\theta}^2 \right\} r dr d\theta \quad (87)$$

$$U_B = \frac{1}{2} \int_0^{2\pi} \int_{r_1}^{r_2} D \left\{ k_r^2 + k_\theta^2 + 2\nu k_r k_\theta + \frac{1}{2}(1-\nu) k_{r\theta}^2 \right\} r dr d\theta \quad (88)$$

where:

$$\left.\begin{aligned} K &= \frac{Et}{1-\nu^2} \\ D &= \frac{Et^3}{12(1-\nu^2)} \end{aligned}\right\} \quad (89)$$

The middle surface strains and the bending distortion for axisymmetric nonlinear deformation of a shallow shell expressed in terms of the middle surface displacements (u, w) are as follows (see page 149 of ref. 41, for example)

$$\left.\begin{aligned} e_r &= \frac{\overline{w}}{R_r} + \frac{1}{2}\left(\frac{d\overline{w}}{dr}\right)^2 + \frac{d\overline{u}}{dr} \\ e_\theta &= \frac{\overline{w}}{R_\theta} - \frac{\overline{u}}{r} \\ e_{r\theta} &= 0 \end{aligned}\right\} \quad (90)$$

$$\left.\begin{aligned} k_r &= -\frac{d^2\bar{w}}{dr^2} \\ k_\theta &= -\frac{1}{r}\frac{d\bar{w}}{dr} \\ k_{r\theta} &= 0 \end{aligned}\right\} \qquad (91)$$

where R_r and R_θ are the principal radii of curvature. The substitution of equations (90) and (91) into equations (87) and (88) and subsequent integration with respect to θ gives the following

$$U_M = \pi \int_{r_1}^{r_2} K \left[\left\{\frac{\bar{w}}{R_r} + \frac{1}{2}\left(\frac{d\bar{w}}{dr}\right)^2 + \frac{d\bar{u}}{dr}\right\}^2 + \left\{\frac{\bar{w}}{R_\theta} - \frac{\bar{u}}{r}\right\}^2\right.$$

$$\left. + 2\nu\left\{\frac{\bar{w}}{R_r} + \frac{1}{2}\left(\frac{d\bar{w}}{dr}\right)^2 + \frac{d\bar{u}}{dr}\right\}\left\{\frac{\bar{w}}{R_\theta} - \frac{\bar{u}}{r}\right\}\right] rdr \qquad (92)$$

$$U_B = \pi \int_{r_1}^{r_2} D\left[\left(\frac{d^2\bar{w}}{dr^2}\right)^2 + \frac{1}{r^2}\left(\frac{d\bar{w}}{dr}\right)^2 + \frac{2\nu}{r}\frac{d^2\bar{w}}{dr^2}\frac{d\bar{w}}{dr}\right] rdr \qquad (93)$$

It is convenient to define the following dimensionless variables

$$\left.\begin{aligned} \rho &= \frac{r}{b} \\ \omega_1 &= \frac{b}{R_r} \\ \omega_2 &= \frac{b}{R_\theta} \\ w &= \frac{\bar{w}}{\bar{l}} \\ u &= \frac{\bar{u}}{\bar{l}} \end{aligned}\right\} \qquad (94)$$

where:

b is the radial distance measured perpendicular to the center line from the center line to the shell edge

ω_1, ω_2 are curvatures

$\bar{l}$ is a reference length

The substitution of relations (94) into equation (92) and (93) leads to the following expression for the strain potential:

$$\bar{U} = \int_0^1 \left\{ k \left[\left(w\,\omega_1 + \frac{l}{2} w'^2 + u' \right)^2 + \left(w\,\omega_2 - \frac{u}{\rho} \right)^2 + 2\nu \left(w\,\omega_1 + \frac{l}{2} w'^2 \right.\right.\right.$$

$$\left.\left.\left. + u' \right) \left(w\,\omega_2 - \frac{u}{\rho} \right) \right] + d \left[w''^2 + \frac{1}{\rho^2} w'^2 + \frac{2\nu}{\rho} w'' w' \right] \right\} \rho d\rho \qquad (95)$$

where:

$$\bar{U} = \frac{U}{\pi D_o l^2}$$

$$l = \frac{\bar{l}}{b}$$

$$k = 12 \frac{E}{E_o} \left(\frac{b}{t}\right)^2 \left(\frac{t}{t_o}\right)$$

$$d = \frac{D}{D_o l^2}$$

$(\ )'$ indicates differentiation with respect to ρ.

D_o, E_o are reference bending stiffness and modulus of elasticity

Potential of Surface Forces

The potential corresponding to the work of the conservative pressure distribution p normal to the surface of the shell is given by expression

$$\Pi_s = -2\pi \int_0^b p\bar{w}r dr \tag{96}$$

In nondimensional form equation (87) becomes

$$\bar{\Pi}_s = -2 \int_0^1 \Omega w \rho d\rho \tag{97}$$

where

$$\overline{\Pi}_s = \frac{\Pi_s}{\pi D_o l^2}$$

$$\Omega = \frac{4}{l^2}\left(\frac{b}{a}\right)^2\left(\frac{b}{t_o}\right)\overline{P}$$

$$\overline{P} = \frac{p}{p_{cr}}$$

and where

$$p_{cr} = \frac{4E_o t_o^{\ 2}}{a^2 m^2}$$

$$m^4 = 12\left(1-\nu^2\right)$$

The parameter p_{cr} is the classical buckling pressure of a complete spherical shell of the same radius of curvature and thickness.

Potential of the Edge Restraint

A generalized potential for arbitrary edge restraints (linear springs) can be written in the following form.

$$\Pi_e = \frac{1}{2}\int_o^{2\pi}\left(q_1\overline{u}_b^{\ 2} + 2q_2\overline{u}_b\overline{w}_b + q_3\overline{w}_b^{\ 2} + q_4\left(\frac{d\overline{w}}{dr}\right)_b^2\right)bd\theta \qquad (98)$$

where q_1, q_2, q_3, and q_4 are appropriate constants (spring rates). After integrating and nondimensionalizing equation (98) can be written as follows:

$$\overline{\Pi}_e = \overline{q}_1 u_b^{\ 2} + 2\overline{q}_2 u_b w_b + \overline{q}_3 w_b^{\ 2} + \overline{q}_4 w_b{}'^2 \qquad (99)$$

where

$$\overline{\Pi}_e = \frac{\Pi_e}{\pi l^2 D_o}$$

$$\overline{q}_i = \frac{q_i b^3}{l^2 D_o} \qquad (i = 1, 2, 3)$$

$$\overline{q}_4 = \frac{q_4 b}{l^2 D_o}$$

Total Potential Energy

The total potential energy Π_t is the sum of the strain potential, the potential of external forces, and the potential of the edge restraint.

Thus:

$$\overline{\Pi}_t = \overline{U} + \overline{\Pi}_e + \overline{\Pi}_s \qquad (100)$$

The substitution of equations (95), (97), and (99) into (100) leads to the following expression for $\overline{\Pi}_t$ in terms of the displacement variables:

$$\overline{\Pi}_t = \int_0^1 \left(k \left[\left\{ w\,\omega_1 + \frac{l}{2} w'^2 + u' \right\}^2 + \left\{ w\,\omega_2 - \frac{u}{\rho} \right\}^2 + 2\nu \left\{ w\,\omega_1 + \frac{l}{2} w'^2 + u' \right\} \left\{ w\,\omega_2 - \frac{u}{\rho} \right\} \right] + d \left[w''^2 + \frac{1}{\rho^2} w'^2 + \frac{2\nu}{\rho} w'' w' \right] - 2\Omega w \right) \rho d\rho + \left[\overline{q}_1 u_b^{\,2} + 2\overline{q}_2 u_b w_b + \overline{q}_3 w_b^{\,2} + \overline{q}_4 w'_b{}^2 \right] \qquad (101)$$

where the subscript b indicates that the displacement is evaluated at the boundary $\rho = 1$.

Numerical Approximation of Potential Functions

The methods of unconstrained function minimization which are presented in chapter VII apply to functions of a finite number of variables. It is necessary therefore to approximate the functional equation (101) by a finite number of unknowns. This is accomplished in the present investigation by approximating the continuous functions (u,w) and the shell properties of N discrete stations along the radius of the shell. These discrete stations are shown in figure 3. The interval between stations Δ is taken to be a constant, and the off-shell station N+1 is added to allow the evaluation of derivatives at station N.

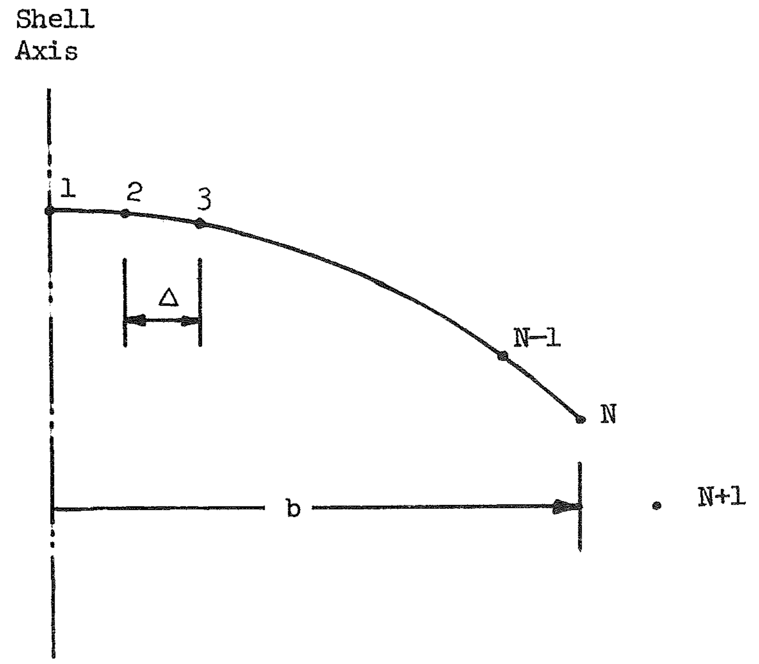

Figure 3.- Meridian of shallow shell showing location of finite-difference stations.

Location of difference approximation.- It will be shown by using a simple example that the principal error term in the numerical approximation of the following integral

$$f(w') = \int_0^1 w'^2 dx \tag{102}$$

is less for a given increment size when the first derivative is evaluated midway between stations.

Consider the following finite-difference approximations for the first derivative at the station i, and midway between the stations i and i+1, respectively

$$w_i' \simeq \frac{1}{2\Delta}\left(w_{i+1} - w_{i-1}\right) \tag{103}$$

$$w'_{i+1/2} \simeq \frac{1}{\Delta}\left(w_{i+1} - w_i\right) \tag{104}$$

The error in each of the approximations given by equations (103) and (104) can be estimated by appropriate Taylor series expansions. The results are as follows

$$w_i' \simeq w_i' + \frac{\Delta^2}{6} w_i''' \tag{105}$$

$$w'_{i+1/2} \simeq w'_{i+1/2} + \frac{1}{48}\Delta^2 w'''_{i+1/2} \tag{106}$$

where terms involving Δ to fourth degree and higher have been neglected.

The principal error term in the numerical integration of equation (102) for each method of specifying the derivative is found by substituting the appropriate relation (105) or (106) into equation (102). The principal error term for each method, neglecting terms of Δ of higher than second degree, is as follows

$$(\text{error})_i \cong \frac{\Delta^2}{3} \int_0^1 w_i{}' w_i{}''' \, dx \tag{107}$$

$$(\text{error})_{i+1/2} \cong \frac{\Delta^2}{24} \int_0^1 w'_{i+1/2} \; w'''_{i+1/2} dx \tag{108}$$

It can be seen by comparing the error terms for the whole station and half-station representations of the first derivative that in this example the half-station method results in a principal error term which is approximately one-eighth of the error associated with the whole station representation.

By proceeding in a manner similar to that presented above, it can be shown that overall numerical error involved in the combined finite-difference approximation and numerical integration is minimized if the following rules are obeyed.

a. Evaluate even derivatives at the stations.

b. Evaluate the odd derivatives midway between stations.

c. If a term in the integral involves the product of an even and an odd derivative, evaluate the product at the location which is appropriate for the highest order derivative appearing in the product.

The numerical approximations of derivatives and function values which are appropriate for a trapezoidal sum approximation of the integral are as follows

$$\left.\begin{aligned}
w_{i+1/2} &\simeq \frac{1}{2}\left(w_{i+1} + w_{i}\right) && + 0\left(\Delta^2\right)\\
w_i' &\simeq \frac{1}{2\Delta}\left(w_{i+1} - w_{i-1}\right) && + 0\left(\Delta^2\right)\\
w'_{i+1/2} &\simeq \frac{1}{\Delta}\left(w_{i+1} - w_{i}\right) && + 0\left(\Delta^2\right)\\
w_i'' &\simeq \frac{1}{\Delta^2}\left(w_{i+1} - 2w_i + w_{i-1}\right) && + 0\left(\Delta^2\right)
\end{aligned}\right\} \qquad (109)$$

The notation $0\left(\Delta^2\right)$ indicates that the principal error term is proportional to Δ^2.

<u>Comparison of gradient and numerical approximation of the Euler-Lagrange equation</u>.- The gradient expression which is derived from the numerical approximation of an integral should reduce to the appropriate Euler-Lagrange equation as the increment size is

reduced to zero. The following example shows that the use of trapezoidal summation to represent an integral together with the numerical approximations given by equation (109) and the rules governing the location at which terms are evaluated leads to a gradient expression which reduces to the Euler-Lagrange equation.

Consider the functional

$$f(w,w') = \int_{0}^{1} \left(w^2 + w'^2\right)dp \qquad (110)$$

subject to the condition that

$$w(0) = w_0$$

$$w(1) = w_1$$

The Euler-Lagrange equation is as follows

$$w - w'' = 0 \qquad (111)$$

By using trapezoidal integration, equation (110) can be approximated by the following summation

$$f(w,w') = \sum_{i=1}^{N} w_i^2 \epsilon_i + \sum_{i=1}^{N-1} w'^2_{i+1/2} \; \epsilon_{i+1/2} \qquad (112)$$

where the even derivatives are evaluated at the stations and odd derivatives are evaluated midway between stations. The quantities

ϵ_i and $\epsilon_{i+1/2}$ are integration factors whose value depends on the location along ρ. These ϵ_i and $\epsilon_{i+1/2}$ are equal to Δ except that $\epsilon_1 = \epsilon_{N-1} = \frac{\Delta}{2}$, $\epsilon_N = \epsilon_{N-1/2} = 0$.

Equation (112) after the substitution of the appropriate expression from equations (109) becomes:

$$f(w,w') = \sum_{i=1}^{N} w_i^2 \epsilon_i + \sum_{i=1}^{N-1} \frac{1}{\Delta^2} \left(w_{i+1} - w_i\right)^2 \epsilon_{i+1/2} \qquad (113)$$

The gradient of the function represented by equation (113) is as follows.

$$\frac{\partial f}{\partial w_i} = 2\Delta \left(w_i - \frac{1}{\Delta^2}\left\{w_{i+1} - 2w_i + w_{i-1}\right\}\right) \qquad (114)$$

The numerical approximation of the Euler-Lagrange equation (111) is as follows.

$$w_i - \frac{1}{\Delta^2}\left(w_{i+1} - 2w_i + w_{i+1}\right) = 0 \qquad (115)$$

Since the vanishing of the gradient components is a necessary condition for a minimum it can readily be seen that equations (114) and (115) are identical. In addition, since the finite-difference approximation of the Euler-Lagrange reduces to the Euler-Lagrange equation as the increment Δ approaches zero it follows that the gradient expression, equation (114) also reduces to the Euler-Lagrange equation.

Numerical approximation of shallow shell potential.- The numerical approximation of the potential function for the shallow shell can be written as follows

$$\overline{\Pi}_t = \overline{\Pi}_L + \overline{\Pi}_{NL} \tag{116}$$

where $\overline{\Pi}_L$ is the portion of $\overline{\Pi}_t$ involving terms in (u,w) of second degree and lower. The terms in (u,w) of greater than second degree are incorporated in $\overline{\Pi}_{NL}$. For convenience $\overline{\Pi}_L$ is called the "linear" potential and $\overline{\Pi}_{NL}$ is called the "nonlinear" potential since the minimization of $\overline{\Pi}_L$ leads to the equations governing linear shell behavior and the inclusion of $\overline{\Pi}_{NL}$ takes into account the nonlinear effects.

The introduction of trapezoidal summations to approximate definite integrals and the use of consistant numerical approximations for derivatives to give a minimum error gives the following expressions for $\overline{\Pi}_L$ and $\overline{\Pi}_{NL}$.

$$\begin{aligned}\overline{\Pi}_L \simeq \sum_{L=1}^{N} \Bigg(\Bigg[& k_{i-1/2} \left(u_{i-1/2}^2 + 2w_{i-1/2}\, u'_{i-1/2} \left(\omega_1 + \nu\, \omega_2\right)_{i-1/2} \right. \\ & \left. + \frac{2\nu}{\rho_{i-1/2}}\, u_{i-1/2}\, u'_{i-1/2} \right) + \frac{d_{i-1/2}}{\rho_{i-1/2}}\, w'^2_{i-1/2} \Bigg] \rho_{i-1/2}\, \epsilon_{i-1/2} \\ & + \Bigg[k_i \left(w_i^2 \left(\omega_1^2 + \omega_2^2 + 2\nu\, \omega_1\omega_2\right)_i + \frac{u_i^2}{\rho_i^2} + \frac{2w_i u_i}{\rho_i} \left(\omega_2 + \nu\, \omega_1\right)_i \right) \\ & + d_i \left(w_i''^2 + \frac{2\nu}{\rho_i}\; w_i' w_i'' \right) \; - 2\Omega_i w_i \Bigg] \rho_i \epsilon_i \Bigg) + \bar{q}_1 u_N^2 + 2\bar{q}_2 u_N w_N \\ & + \bar{q}_3 w_N^2 + \bar{q}_4 w'^2_N \end{aligned} \qquad (117)$$

$$\begin{aligned}\overline{\Pi}_{NL} \simeq \sum_{L=1}^{N} k_{i-1}\, l \Bigg\{ & \frac{l w^4_{i-1/2}}{4} + w'^2_{i-1/2} \Bigg(w_{i-1/2} \left(\omega_1 + \nu\, \omega_2\right) + u'_{i-1/2} \\ & + \frac{\nu u_{i-1/2}}{\rho_{i-1/2}} \Bigg) \Bigg\} \rho_{i-1/2}\, \epsilon_{i-1/2} \end{aligned} \qquad (118)$$

In the following the relations (117) and (118) will be considered to be equalities rather than approximations.

Closure Conditions

The expressions for the circumferential midplane strain ϵ_θ and the midplane bending curvature k_θ equations (90) and (91), respectively, are seen to have a first-order pole at the center of the shell. In order for the strain and bending curvature expressions to be finite as $r \to 0$ it is necessary that $\bar{u}$ and $\frac{\partial \bar{w}}{\partial r}$ approach zero at the same rate as r; thus, at $r = 0$ the following conditions must hold

$$\bar{u} = 0$$

$$\frac{\partial \bar{w}}{\partial r} = 0$$

Or, in terms of the nondimensional variables;

$$u = 0$$

$$w' = 0$$

The appropriate strain and bending curvature expressions at the center of the shell are as follows:

$$\left.\begin{aligned} \epsilon_\theta &= \frac{\bar{w}}{R_\theta} - \bar{u}' \\ k_\theta &= - \frac{d^2 w}{dr^2} \end{aligned}\right\} \qquad (119)$$

Gradient Components

The gradient components are found by differentiating the energy expression (116) with respect to the nodal displacements. Differentiation of equation (117) with respect to the nodal displacements u_i, w_i gives the following gradients of the linear potential $\overline{\Pi}_L$

$$
\begin{aligned}
\frac{\partial \overline{\Pi}_L}{\partial w_i} = {} & \epsilon_{i+1/2}\, \rho_{i+1/2} \Bigg[k_{i+1/2} \Big\{ u'_{i+1/2} \left(\omega_1 + \nu\omega_2\right)_{i+1/2} \Big\} \\
& + 2\, \frac{d_{i+1/2}}{\rho^2_{i+1/2}}\, w'_{i+1/2} \left(- \frac{1}{\Delta} \right) \Bigg] + \epsilon_{i-1/2}\, \rho_{i-1/2} \Bigg[k_{i-1/2} \Big\{ u'_{i-1/2} \big(\omega_1 \\
& + \nu\, \omega_2 \big)_{i-1/2} \Big\} + 2\, \frac{d_{i-1/2}}{\rho^2_{i-1/2}}\, w'_{i-1/2} \left(\frac{1}{\Delta} \right) \Bigg] + \epsilon_i \rho_i \Bigg[b_i \Big\{ 2w_i \big(\omega_1^{\,2} + \omega_2^{\,2} \\
& + 2\nu\omega_1\omega_2 \big)_i + 2\, \frac{u_i}{\rho_i} \left(\omega_2 + \nu\omega_1 \right)_i \Big\} + d_i \Big\{ 2w_i'' \left(- \frac{2}{\Delta^2} \right) + \frac{2\nu}{\rho_i}\, w_i' \left(- \frac{2}{\Delta^2} \right) \Big\} \\
& - 2\Omega_i \Bigg] + \epsilon_{i+1}\, \rho_{i+1}\, d_{i+1} \Big\{ 2w''_{i+1} \left(\frac{1}{\Delta^2} \right) + \frac{2\nu}{\rho_{i+1}}\, \frac{w'_{i+1/2}}{\Delta^2} \Big\} \\
& + \epsilon_{i-1}\, \rho_{i-1}\, d_{i-1} \Big\{ 2w''_{i-1} \left(\frac{1}{\Delta^2} \right) + \frac{2\nu}{\rho_{i-1}}\, \frac{w'_{i-1/2}}{\Delta^2} \Big\} \\
& + 2\overline{q}_2\, u_{N-1}\, \delta_{i,N-1} + 2\overline{q}_3 w_{N-1}\, \delta_{i,N-1} + \frac{\overline{q}_4}{\Delta} \left(w_N\, \delta_{i,N} - w_{N-2}\, \delta_{i,N-2} \right)
\end{aligned}
\tag{120}
$$

$$(i = 1, 2, \ldots, N+1)$$

$$\begin{aligned}
\frac{\partial \overline{\Pi}_L}{\partial u_i} = {} & \epsilon_{i+1/2}\, \rho_{i+1/2}\, k_{i+1/2} \Bigg\{ 2u'_{i+1/2} \left(- \frac{1}{\Delta}\right) + 2w_{i+1/2} \left(- \frac{1}{\Delta}\right) (\omega_1 \\
& + \nu \omega_2)_{i+1/2} + \frac{2\nu}{\rho_{i+1/2}} \left(u_{i+1/2} \left(- \frac{1}{\Delta}\right) + u'_{i+1/2} \left(\frac{1}{2}\right)\right)\Bigg\} \\
& + \epsilon_{i-1/2}\, \rho_{i-1/2}\, k_{i-1/2} \Bigg\{ 2u'_{i-1/2} \left(\frac{1}{\Delta}\right) + 2w_{i-1/2} \left(\frac{1}{\Delta}\right) (\omega_1 \\
& + \nu\omega_2)_{i-1/2} + \frac{2\nu}{\rho_{i-1/2}} \left(u_{i-1/2} \left(\frac{1}{\Delta}\right) + \frac{1}{2}\, u'_{i-1/2}\right)\Bigg\} + \epsilon_i k_i \rho_i \Bigg\{ 2 \frac{u_i}{\rho_i^2} \\
& + 2\, \frac{w_i}{\rho_i} \left(\omega_2 + \nu\omega_1\right)_i \Bigg\} + \left(2\overline{q}_1\, u_{N-1} + 2\overline{q}_2\, w_{N-1}\right) \delta_{i,N-1}
\end{aligned} \qquad (121)$$

$$(i = 1, 2, \ldots, N+1)$$

where ϵ_i and $\epsilon_{i+1/2}$ are integrating factors having values as follows:

$$\begin{aligned}
\epsilon_{N+1/2} &= 0 & & \\
\epsilon_{i+1/2} &= \Delta & & i = 1, 2, \ldots, N \\
\epsilon_i &= \Delta & & i = 2, 3, \ldots, N-1 \\
\epsilon_1 &= 0 & & \\
\epsilon_{N+1} &= 0 & & \\
\epsilon_N &= \frac{\Delta}{2} & &
\end{aligned}$$

and N is the number of stations on the shell. An off-shell $N+1$ station is considered so that the derivative of (u, w) may be evaluated at the edge. The quantity δ_{ij} is the kronecker delta and is defined as follows:

$$\delta_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

In a similar manner contribution of the nonlinear potential to the gradient components is as follows

$$\begin{aligned}
\frac{\partial \overline{\Pi}_{NL}}{\partial w_i} = {} & k_{i+1/2}\ \rho_{i+1/2}\ \epsilon_{1+1/2}\ l \left[l w'^3_{i+1/2} \left(-\frac{1}{\Delta}\right) \right. \\
& + 2w'_{i+1/2} \left(-\frac{1}{\Delta}\right) \left\{ w_{i+1/2} \left(\omega_1 + \nu\omega_2\right)_{i+1/2} + u'_{i+1/2} + \nu \frac{u_{i+1/2}}{\rho_{i+1/2}} \right\} \\
& \left. + \frac{w'^2_{i+1/2}}{2} \left(\omega_1 + \nu\omega_2\right)_{i+1/2} \right] + k_{i-1/2}\ \rho_{i-1/2}\ \epsilon_{i-1/2}\ l \left[l w'^3_{i-1/2} \left(\frac{1}{\Delta}\right) \right. \\
& + 2w_{i-1/2} \left(\frac{1}{\Delta}\right) \left\{ w_{i-1/2} \left(\omega_1 + \nu\omega_2\right)_{i-1/2} + u'_{i-1/2} + \nu \frac{u_{i-1/2}}{\rho_{i-1/2}} \right\} \\
& \left. + \frac{w'^2_{i-1/2}}{2}\ \omega_1 + \nu\omega_2\ {}_{i-1/2} \right] \qquad (122)
\end{aligned}$$

$$\frac{\partial \overline{\Pi}_{NL}}{\partial u_i} = k_{i+1/2}\, \rho_{i+1/2}\, \epsilon_{i+1/2}\, l \left[w'^2_{i+1/2} \left(- \frac{1}{\Delta} + \frac{\nu}{2\rho_{i+1/2}} \right) \right]$$

$$+ k_{i-1/2}\, \rho_{i-1/2}\, \epsilon_{i-1/2}\, l \left[w'^2_{i-1/2} \left(\frac{1}{\Delta} + \frac{\nu}{2\rho_{i-1/2}} \right) \right] \qquad (123)$$

where

$$(i = 1, 2, \ . \ . \ ., N+1)$$

It is interesting to note the following points with regard to the derivation of equations (120) through (123).

1. Only the shell closure conditions necessary to insure finite strain and bending distortion are specified. This is in contrast to the case where a comparable set of equations is derived by substituting finite-difference expressions in the Euler-Lagrange equations. In the latter case it is also necessary to specify the conditions necessary to guarantee that no concentrated forces exist at the shell apex. In deriving the set of equations directly from the energy expression the latter conditions are automatically satisfied.

2. The set of equations represented by the gradient components is symmetric as a consequence of the quadratic potential function. This is in contrast to the set of finite-difference equations normally derived from the Euler-Lagrange equations which are associated with the potential function (eq. (101)). Symmetry conditions are useful in reducing the number of numerical operations

necessary to determine a solution and in reducing storage requirements in the computer core.

3. There is some indication the inherent discretization error associated with the set of equations derived directly from the potential is less than the corresponding error in the set derived from the Euler-Lagrange equations. This conclusion is drawn based on the results presented by Cyrus (ref. 42) where the error involved in two finite-differences approximations was studied. It was shown that, in general, the system corresponding to the set derived directly from the energy expression has less error, on the average, than the set obtained by approximating the Euler-Lagrange equations.

Second Derivative of the Potential Function

The second derivatives of the potential function which are required in the Newton-Raphson procedure are given by the following expressions.

$$\frac{\partial^2 \bar{\Pi}_L}{\partial w_i^{\,2}} = \frac{2\bar{d}_{i+1/2}}{\Delta^2 \rho^2_{\;i+1/2}} + \frac{2\bar{d}_{i-1/2}}{\Delta^2 \rho^2_{\;i-1/2}} + 2\bar{k}_i \left(\omega_1^{\,2} + \omega_2^{\,2} + 2\nu\omega_1\omega_2\right)_i$$

$$+ \frac{8\bar{d}_i}{\Delta^4} + \bar{d}_{i+1}\left(\frac{2}{\Delta^4} - \frac{2\nu}{\rho_{i+1}\Delta^3}\right) + \bar{d}_i\left(\frac{2}{\Delta^4} + \frac{2\nu}{\rho_{i-1}\Delta^3}\right)$$

$$+ 2\bar{q}_3\, \delta_{i,N-1} + \frac{\bar{q}_4}{\Delta}\left(\delta_{i,N} - \delta_{i,N-2}\right) \qquad (124)$$

$$\frac{\partial^2 \bar{\Pi}_L}{\partial w_i \partial w_{i-1}} = \bar{d}_i \left(- \frac{4}{\Delta^4} + \frac{2\nu}{\rho_i \Delta^3} \right) - \frac{2\bar{d}_{i-1/2}}{\rho^2_{i-1/2}\Delta^2} - \bar{d}_{i-1} \left(\frac{4}{\Delta^4} + \frac{2\nu}{\rho_{i-1}\Delta^3} \right) \tag{125}$$

$$\frac{\partial^2 \bar{\Pi}_L}{\partial w_i \; \partial w_{i+1}} = - \bar{d}_i \left(\frac{4}{\Delta^4} + \frac{2\nu}{\rho_i \Delta^3} \right) - \frac{2\bar{d}_{i+1/2}}{\rho^2_{i+1/2}\Delta^2} + \bar{d}_{i+1} \left(- \frac{4}{\Delta^4} + \frac{2\nu}{\rho_{i+1}\Delta^3} \right) \tag{126}$$

$$\frac{\partial^2 \bar{\Pi}_L}{\partial w_i \partial w_{i-2}} = \frac{2\bar{d}_{i-1}}{\Delta^4} \tag{127}$$

$$\frac{\partial^2 \bar{\Pi}_L}{\partial w_i \partial w_{i+2}} = \frac{2\bar{d}_{i+1}}{\Delta^4} \tag{128}$$

$$\frac{\partial^2 \bar{\Pi}_L}{\partial w_i \partial u_i} = - \bar{k}_{i+1/2} \; \frac{\left(\omega_1 + \nu \omega_2\right)_{i+1/2}}{\Delta} + \bar{k}_{i-1/2} \; \frac{\left(\omega_1 + \nu \omega_2\right)_{i-1/2}}{\Delta}$$

$$+ \frac{2\bar{k}_i}{\rho_i} \left(\omega_2 + \nu \omega_1\right)_i + 2\bar{q}_2 \; \delta_{i,N-1} \tag{129}$$

$$\frac{\partial^2 \bar{\Pi}_L}{\partial u_i \partial w_{i-1}} = \bar{k}_{i-1/2} \, \frac{\left(\omega_1 + \nu \omega_2\right)_{i-1/2}}{\Delta} \tag{130}$$

$$\frac{\partial^2 \bar{\Pi}_L}{\partial u_i \partial w_{i+1}} = - \bar{k}_{i+1/2} \, \frac{\left(\omega_1 + \nu \omega_2\right)_{i+1/2}}{\Delta} \tag{131}$$

$$\frac{\partial^2 \bar{\Pi}_L}{\partial u_i{}^2} = \bar{k}_{i+1/2}\left(\frac{2}{\Delta^2} - \frac{2\nu}{\rho_{i+1/2}\Delta}\right) + \bar{k}_{i-1/2}\left(\frac{2}{\Delta^2} + \frac{2\nu}{\rho_{i-1/2}\Delta}\right) + \frac{2\bar{k}_i}{\rho_i{}^2}$$

$$+ 2\bar{q}_1\ \delta_{i,N-1} \qquad (132)$$

$$\frac{\partial^2 \bar{\Pi}_L}{\partial u_i \partial u_{i-1}} = -\frac{2\bar{k}_{i-1/2}}{\Delta^2} \qquad (133)$$

$$\frac{\partial^2 \bar{\Pi}_L}{\partial u_i \partial u_{i+1}} = -\frac{2\bar{k}_{i+1/2}}{\Delta^2} \qquad (134)$$

where

$$\bar{k}_i = k_i \rho_i \epsilon_i$$

$$\bar{d}_i = d_i \rho_i \epsilon_i$$

Similary, for second derivatives of the nonlinear potential $\bar{\Pi}_{NL}$ are as follows:

$$\begin{aligned}\frac{\partial^2 \bar{\Pi}_{NL}}{\partial w_i^{\,2}} = \bar{k}_{i+1/2} \Bigg[& 3l\, w'^2_{i+1/2} \left(\frac{1}{\Delta}\right)^2 + \frac{2}{\Delta^2} \Big\{ w_{i+1/2} \left(\omega_1 + \nu\omega_2\right)_{i+1/2} \\ & + u'_{i+1/2} + \frac{\nu u_{i+1/2}}{\rho_{i+1/2}} \Big\} - \frac{2}{\Delta}\, w'_{i+1/2} \left(\omega_1 + \nu\omega_2\right)_{i+1/2} \Bigg] \\ - \bar{k}_{i-1/2} \Bigg[& 3l\, w'^2_{i-1/2} \left(\frac{1}{\Delta^2}\right) + \frac{2}{\Delta^2} \Big\{ w_{i-1/2} \left(\omega_1 + \nu\omega_2\right)_{1-1/2} \\ & + u'_{i-1/2} + \frac{\nu u_{i-1/2}}{\rho_{i-1/2}} \Big\} + \frac{2}{\Delta}\, w'_{i-1/2} \left(\omega_1 + \nu\omega_2\right)_{i-1/2} \Bigg] \end{aligned} \qquad (135)$$

$$\begin{aligned}\frac{\partial^2 \bar{\Pi}_{NL}}{\partial w_i \partial w_{i-1}} = \bar{k}_{i-1} \Bigg[& - \frac{3l}{\Delta^2}\, w'^2_{i-1/2} - \frac{2}{\Delta^2} \Big\{ w_{i-1/2} \left(\omega_1 + \nu\omega_2\right)_{i-1/2} \\ & + u'_{i-1/2} + \frac{\nu u_{i-1/2}}{\rho_{i-1/2}} \Big\} \Bigg] \end{aligned} \qquad (136)$$

$$\begin{aligned}\frac{\partial^2 \bar{\Pi}_{NL}}{\partial w_i \partial w_{i+1}} = \bar{k}_{i+1} \Bigg[& - \frac{3l}{\Delta^2}\, w'^2_{i+1/2} - \frac{2}{\Delta^2} \Big\{ w_{i+1/2} \left(\omega_1 + \nu\omega_2\right)_{i+1/2} \\ & + u'_{i+1/2} + \frac{\nu u_{i+1/2}}{\rho_{i+1/2}} \Big\} \Bigg] \end{aligned} \qquad (137)$$

$$\frac{\partial^2 \bar{\Pi}_{NL}}{\partial \omega_i \partial u_{i-1}} = \bar{k}_{i-1/2} \left[\frac{2}{\Delta} w'_{i-1/2} \left\{ -\frac{1}{\Delta} + \frac{\nu}{2\rho_{i-1/2}} \right\} \right] \tag{138}$$

$$\frac{\partial^2 \bar{\Pi}_{NL}}{\partial \omega_i \partial u_i} = \bar{k}_{i+1/2} \left[-\frac{2}{\Delta} w'_{i+1/2} \left\{ -\frac{1}{\Delta} + \frac{\nu}{2\rho_{i+1/2}} \right\} \right] + \bar{k}_{i-1/2} \left[\frac{2w'_{i-1/2}}{\Delta} \left\{ \frac{1}{\Delta} + \frac{\nu}{2\rho_{i-1/2}} \right\} \right] \tag{139}$$

$$\frac{\partial^2 \bar{\Pi}_{NL}}{\partial \omega_i \partial u_{i+1}} = \bar{k}_{i+1/2} \left[-\frac{2}{\Delta} w'_{i+1/2} \left\{ \frac{1}{\Delta} + \frac{\nu}{2\rho_{i+1/2}} \right\} \right] \tag{140}$$

where

$$(i = 1, 2, \ldots, N+1)$$

IX. EVALUATION OF UNCONSTRAINED MINIMIZATION TECHNIQUES

Before discussing the results for the nonlinear shallow shell, which are presented in chapter X,it is appropriate to summarize and evaluate the algorithms presented in chapter VII.

Minimization of a Quadratic Object Function

The conjugate gradient method as presented by Hestenes (ref. 29) and Beckman (ref. 30) is in theory an elegant algorithm for finding the solution of a linear set of equations, or equivalently of finding the minimum of a quadratic function. The method is advertised to be quadratically convergent so that for a function of n-unknowns, the solution, or minimum, is located in at most n-steps neglecting round-off error.

A computer program applying the conjugate gradient method for finding the solution of a linear set of equations was written using single precision floating point arithmetic. The set of equations associated with the linear bending of a circular plate was then solved using 10 equally spaced increments along the meridian on a computer which uses 8 equally significant figures in numerical calculations. The iteration process converged; however, on the order of 400 iterations were required for a problem involving only 22 unknowns indicating that round-off errors were prominent. In order to reduce the effect of round-off error the program was modified to use double precision arithmetic (16 significant figures).

The double precision program converged in approximately 40 iterations indicating that round-off error was still dominant. The conclusion was reached that in spite of the attractiveness of using the conjugate gradient method for the solution of linear problems, the round-off error and the necessity of using double precision arithmetic make the method noncompetitive with a standard matrix inversion subroutine such as Jordan's method.

White (ref. 43) investigated several solution techniques in order to determine, at least qualitatively, the best digital computer solution method for the finite-difference equations. Four methods were considered; conjugate gradients, Gauss-Seidel iteration, accelerated Gauss-Seidel iteration,and Gaussian elimination. White reached the same conclusions with regard to conjugate gradient methods as are reached in the present investigation. It is interesting to note that White found that, for the class of problems considered, Gaussian elimination gives the best solution and uses the least computer time. This same conclusion was reached by Beckman, who points out that experience in applying the conjugate gradient method to some large linear systems indicates that the method compares unfavorably with Gaussian elimination. Beckman further points out that "the elimination method can be applied using double precision arithmetic to a linear system with few zero coefficients without involving more elementary arithmetic (single precision) operations than the conjugate gradient method." In view of the experience of the several investigators, it appears that the conjugate gradient method should not be used to solve a set of linear equations.

The variable metric method was also programed to find the solution of a linear set of equations using single precision arthmetic operations. The method converges in the theoretical n-cycles, without apparent round-off error. However, since the standard library program for matrix inversion gives results which are as accurate as the variable metric method in what appears to be less computer time, the standard Langley Computer Library subroutine for matrix inversion has been used in the present investigation, and appears to be very efficient for the number of variables considered.

It would appear that for a large set of finite-difference equations that some form of Gaussian elimination which takes advantage of the banded matrix would be appropriate. However, because the number of unknowns in the present investigation is not large, a general computer program for solving the linear set of equations using Gaussian elimination has not been prepared.

Minimization of Nonquadratic Function

Three methods are presented in chapter VII which have been used by various investigators to minimize nonquadratic object functions. It has already been stated previously that the conjugate gradient method suffers from round-off error when applied to the minimization of a quadratic function. Thus, while the algorithm of conjugate gradients might be expected to be more efficient than the method of steepest descent, it would not be expected to converge as well as theory might indicate. The method employing the variable metric on the other hand could be expected to locate the minimum in a much more efficient manner than the conjugate gradient method because round-off errors are not significant.

Each of these methods has been applied to the problem of locating the unconstrained minimum of the function

$$f\left(x_1,x_2\right) = 100\left(x_2-x_1^{\,2}\right)^2 + \left(1-x_1\right)^2 \qquad (141)$$

starting from the point $\left(x_1,x_2\right) = (-1.2,\ 1.0)$ (see ref. 31 and 34). The results tend to verify the general conclusions drawn above. The minimum, which is located at $x_1,x_2 = (1,1)$ is found in 18 iterations by the variable metric method and in 27 iterations by the conjugate gradient method. The value of the function associated with each method is 10^{-8}. It is interesting to note that the steepest descent method has only reduced the function value from 24.200 to 2.18 in 27 iterations, indicating that while the variable metric technique is the more powerful method, the method of conjugate gradient with round-off error is superior to the method of steepest descent.

The variable metric method was evaluated further by Fletcher and Powell to determine if the variable metric method is suitable for finding the minimum of a function of a large number of variables. Nonlinear functions of up to 100 variables were considered with extremely good results. A set of random numbers was chosen as the starting point and for the case of 100 unknowns a solution of a set of nonlinear equations was found in 162 iterations which required 318 function evaluations. Cubic interpretation in the one-dimensional search was used. The variable metric method has been applied to the minimization of sample functions in the course of the present investigation and the same fast convergence has been found, and confirms the following statement by Fletcher and Powell "the method (variable metric) is probably the most powerful general procedure for finding a local minimum which is known at the present time."

While the variable metric method is a very powerful minimization technique, the necessity of storing and manipulating a square matrix of order n in the course of the minimization process was judged unacceptable for the class of problems that is contemplated. Thus attention was focused on the conjugate gradient and Newton-Raphson techniques.

Early investigation of the conjugate gradient technique indicated that an unacceptably large number of iterations would be required to locate the unconstrained minimum of the shallow shell potential function. In the present investigation the notion of restarting the process from a steepest descent every $(n+1)$ iteration as recommended by Fletcher and Reeves was adopted without success. The difficulty is that the process should be allowed to continue for the number of iterations that would be required to locate the minimum of a quadratic function. However, because of round-off error in the calculations this number was not known a priori. During the course of investigating various techniques for speeding convergence, the minimization process was found to be significantly improved if the one-dimensional search interpolation procedure was repeated until the angle between p_i^{α} and $g_i^{\alpha+1}$ was within 0.001 radians of the theoretical $\pi/2$ radians, or until no improvement could be made. It was found, in general, that no significant improvement was made after the second interpolation.

The algorithm which was finally chosen for locating the minimum of the object function of the shallow shell was a hybrid. It combined the Newton-Raphson method as described in chapter VII and the conjugate gradient method. This combination was chosen for the following reasons:

1. The matrix $\frac{\partial^2 f}{\partial x_i \partial x_j}$ is symmetric and has a narrow band of nonzero terms. The error vector e_j^a equation (68) can thus be found accurately and quickly by using a Gaussian elimination technique. Only the nonzero terms need to be stored in the computer.

2. The Newton-Raphson method converges quickly for those cases where the necessary conditions for a contraction mapping are satisfied. For a contraction mapping the inequality (71) must be satisfied.

3. For those cases where the inequality (71) is not satisfied, the conjugate gradient technique is an acceptable method for moving from the current point to another point where the inequality will be satisifed. The conjugate gradient procedure is acceptable because a relatively few iterations are usually required in practice and the algorithm utilized a minimum of available core storage.

X. DISCUSSION OF RESULTS FOR SHALLOW SHELL

Description of Program Logic

During the evaluation phase of this investigation a combination of the Newton-Raphson procedure and the conjugate gradient method was found to be well suited to the solution of nonlinear equations. The details of how this minimization procedure was utilized in obtaining the buckling loads and deflections for shallow shells of revolution is described in this section.

The load-deflection curve for a uniformly loaded shallow shell having clamped edges has the general form shown in figure 4. (See Thurston (ref. 22), for instance.) The deflection is found to increase monotonically for $\mu^2 < 11$, where $\mu^2 = \frac{m^2b^2}{at}$, $m^4 = 12\left(1-\nu^2\right)$ and other quantities have been previously defined. For $\mu^2 > 11$ the curve has two horizontal tangents labeled U and L in figure 4. In the prebuckled range, OU, the Newton-Raphson procedure converges strongly when the linear solution is used as the initial state point. For loads greater than that associated with the horizontal tangent U the Newton-Raphson procedure did not converge when the most recent converged state point or the linear solution is used as an initial state.

In order to determine a solution in the post buckled range LN a very crude initial estimate of the postbuckled deflection shape was made by assuming that the center deflection is -2H, the edge deflection is zero, and that the deflection varies linearly

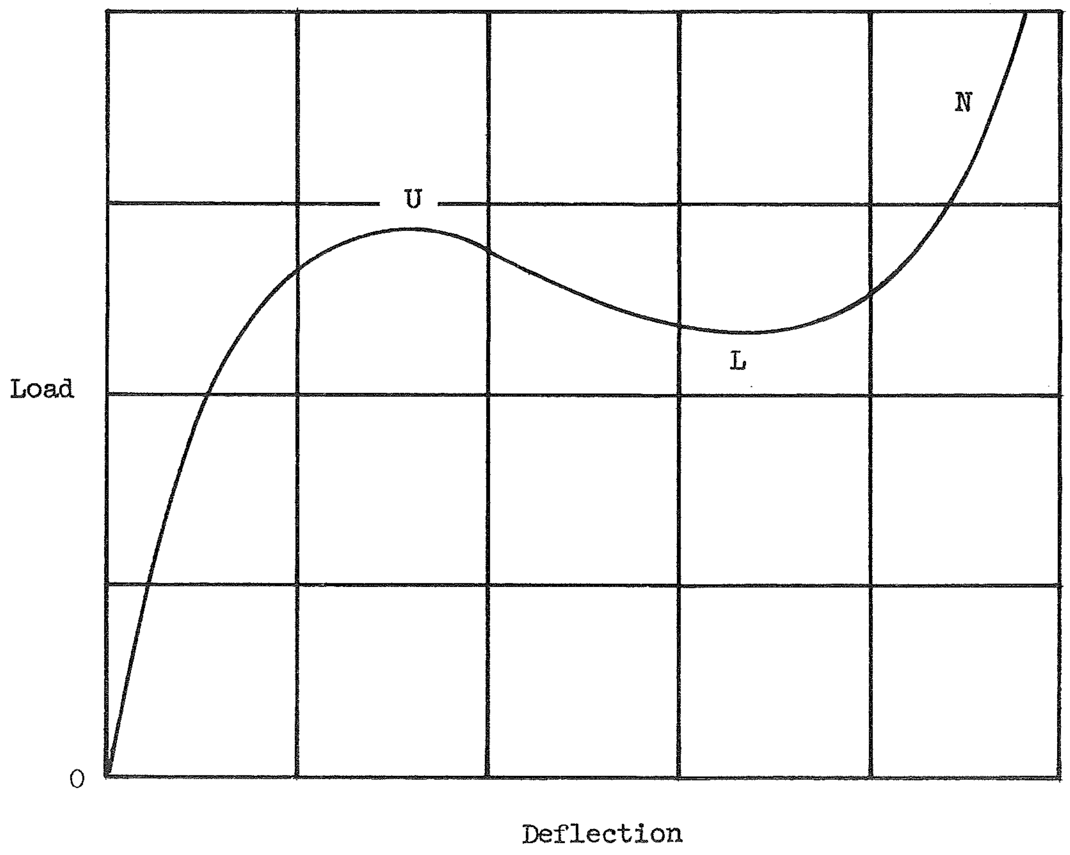

Figure 4.- Load deflection curve for a uniformly loaded shallow spherical shell, typical of the results presented by Thurston (ref. 22).

between these extremes. It is unusual if the Newton-Raphson procedure converges when this initial estimate is used. If the Newton-Raphson procedure does not converge, the potential is minimized using the conjugate gradient procedure starting from the crude estimate of the post buckled shape. The Newton-Raphson procedure is periodically applied using the then current state point to determine if the Newton-Raphson procedure will converge. Experience has shown that a very few cycles (on the order of $3n$ or less when n is the number of unknowns) is sufficient to reach a state point from which the Newton-Raphson procedure will converge. Once a solution in the post buckled range is found the curve NL can be defined by incrementing the load, where the initial state associated with each load is taken to be the last converged result.

The curve UL represents unstable equilibrium conditions for the shallow shell. These states have not been determined in the present investigation because they are only of academic interest and are not generally realizable and the expense of computational effort necessary to determine these states does not appear warranted.

Two convergence tests are made in the Newton-Raphson procedure. The first determines that the function is being minimized. In order that the minimization process be taking place it is necessary that

$$f^{a+1} < f^{a} \tag{142}$$

where f^{a} is the value of the potential after the a<u>th</u> application of the Newton-Raphson procedure.

If the inequality represented by equation (142) is satisfied then the error vector e_j^a (see equation (68)) is compared to the vector x_j^a. If the norm of e_j^a, $\| e^a \|$, is less than $\delta \| x^a \|$, where δ is a preassigned small number, then the estimate of the location of the minimum

$$\bar{h}_j^a = x_j^a + e_j^a \tag{143}$$

is accepted as the converged value. In the present investigation δ is taken as 0.01.

If the inequality (142) is not satisfied, then the load increment is reduced and the Newton-Raphson procedure is again applied. If two successive reductions of the load increment do not lead to convergence in the sense of relation (142) then it is assumed that the load associated with the last converged result is the buckling load. In the present investigation the new load increment is taken as 1/5 of the current load increment so that the buckling load is found to within 0.025 $\Delta\bar{P}$ where $\Delta\bar{P}$ is the magnitude of the load increment. When the Newton-Raphson procedure is reinitiated after reducing the load increment, the last converged solution is used as an initial approximation of the location of the minimum.

The computer program which implements the logic described above and which incorporates the potential and gradients for the shallow shell is presented in the appendix.

Numerical Results

Huang (ref. 6) has shown that asymmetric behavior governs the buckling of a uniformly loaded spherical cap for $\mu > 6$. The present investigation is therefore restricted to the study of shells with $\mu \geq 6$ since axisymmetric results for $\mu > 6$ have little practical value.

The investigations of the nonlinear behavior of uniformly loaded spherical caps which have been presented in the literature have considered shells with either fully clamped or completely unrestrained edges. Results were obtained in this study which allow an investigation of the influence of arbitrary edge conditions on the buckling load to be carried out. These results were obtained for a variation in meridional restraint as well as rotational restraint.

Effect of meridional edge restraint.- Calculations were made to determine the influence of meridional restraint on the buckling behavior of shallow spherical shells. Plots of the buckling load $\overline{P}_{cr}$ for various values of the inplane restraint parameter $\bar{q}_1$ are shown in figures 5, 6, and 7 for geometric parameters $\mu = 4, 5,$ and $6,$ respectively. The calculations were carried out for a shallow spherical cap with the edge fully restrained against rotation $\left(\bar{q}_4 = 10^5\right)$ and no cross-coupling between normal and meridional displacements at the edge $\left(\bar{q}_2 = 0\right)$ and with $\nu^2 = 0.1$, $b/t = 100$. The plots show that the buckling load is essentially independent of the inplane restraint for $\bar{q}_1 < 10^3$ and $\bar{q}_1 > 10^7$.

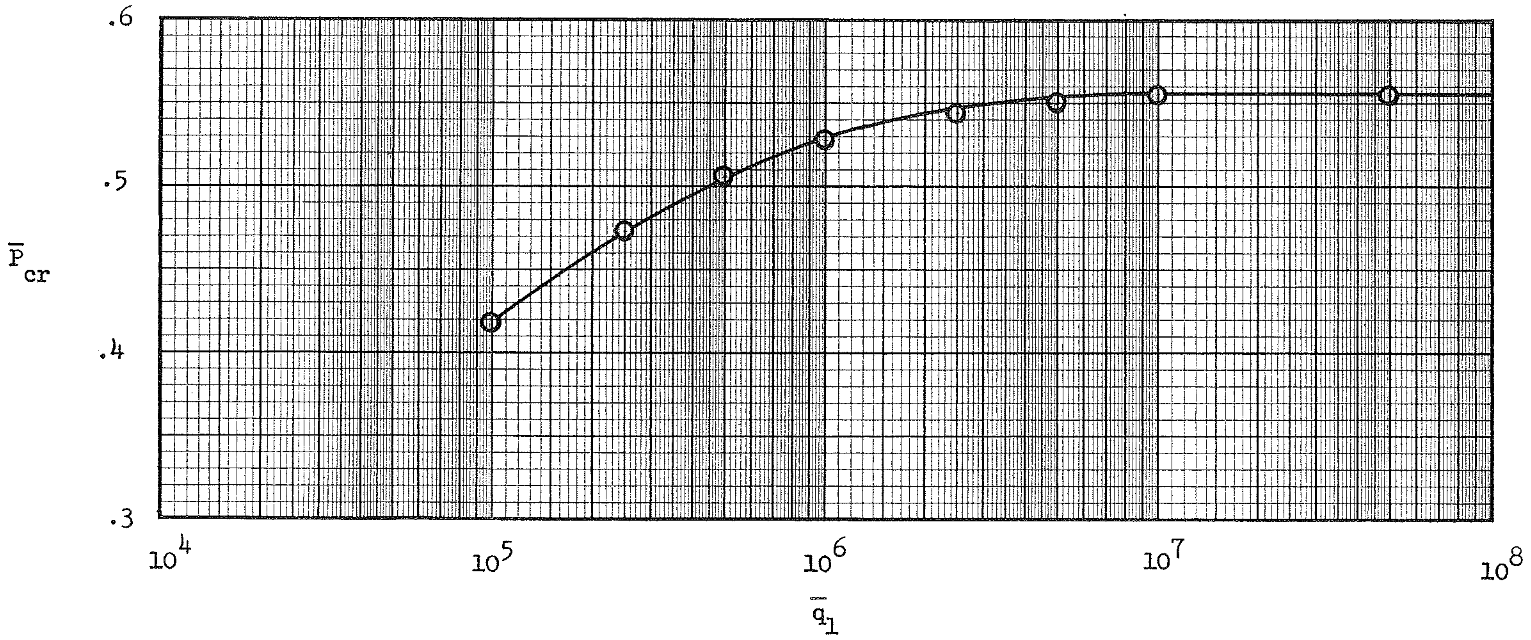

Figure 5.- Buckling load $\bar{P}_{cr}$ as a function of inplane restraint $\bar{q}_1$ for shell parameter $\mu = 4$, $\nu^2 = 0.1$, $b/t = 100$, $\bar{q}_2 = 0$, $\bar{q}_3 = 10^5$, $\bar{q}_4 = 10^5$.

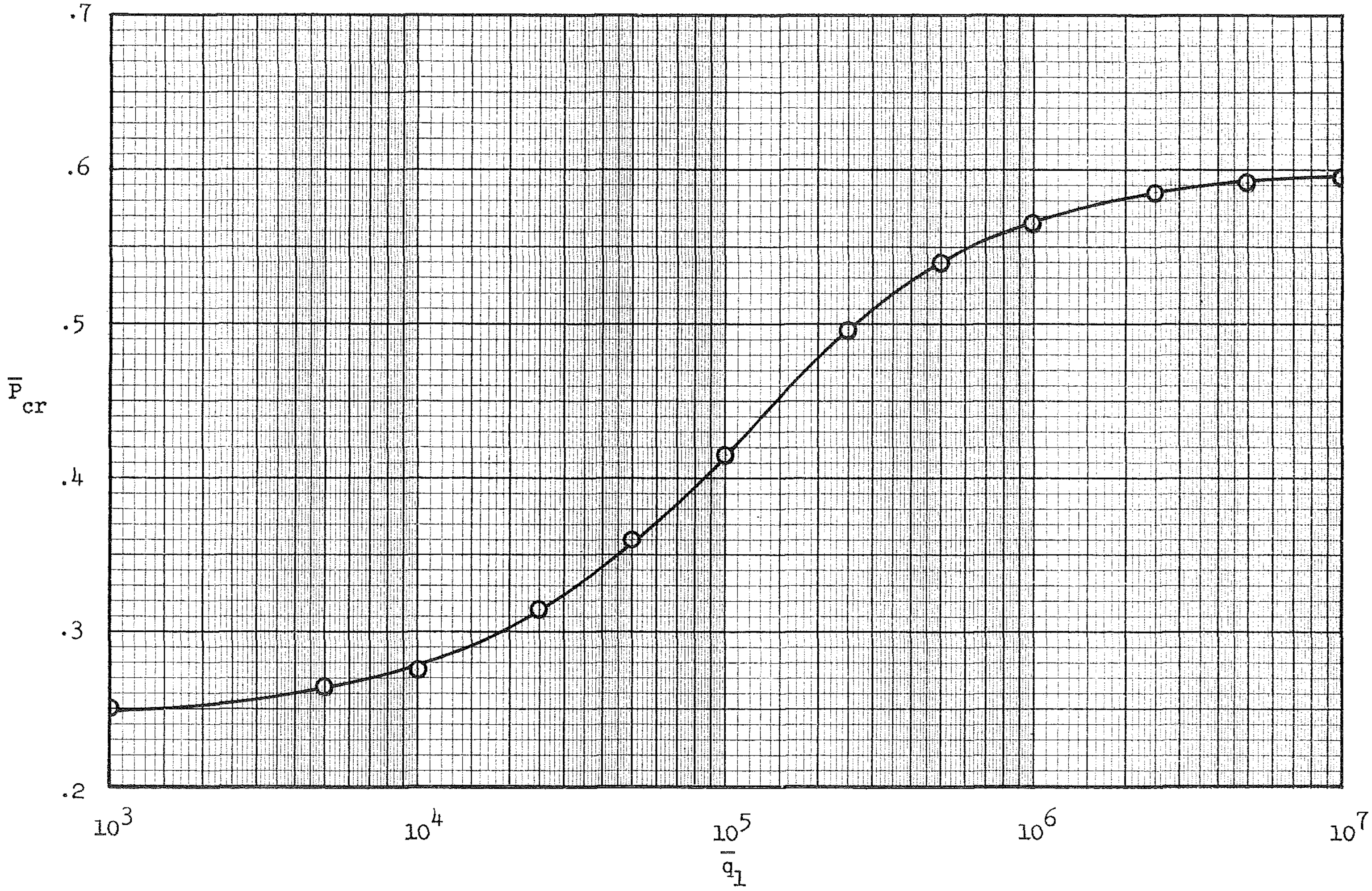

Figure 6.- Buckling load $\bar{P}_{cr}$ as a function of inplane restraint $\bar{q}_1$ for shell parameter $\mu = 5$, $\nu^2 = 0.1$, $b/t = 100$, $\bar{q}_2 = 0$, $\bar{q}_3 = 10^5$, $\bar{q}_4 = 10^5$.

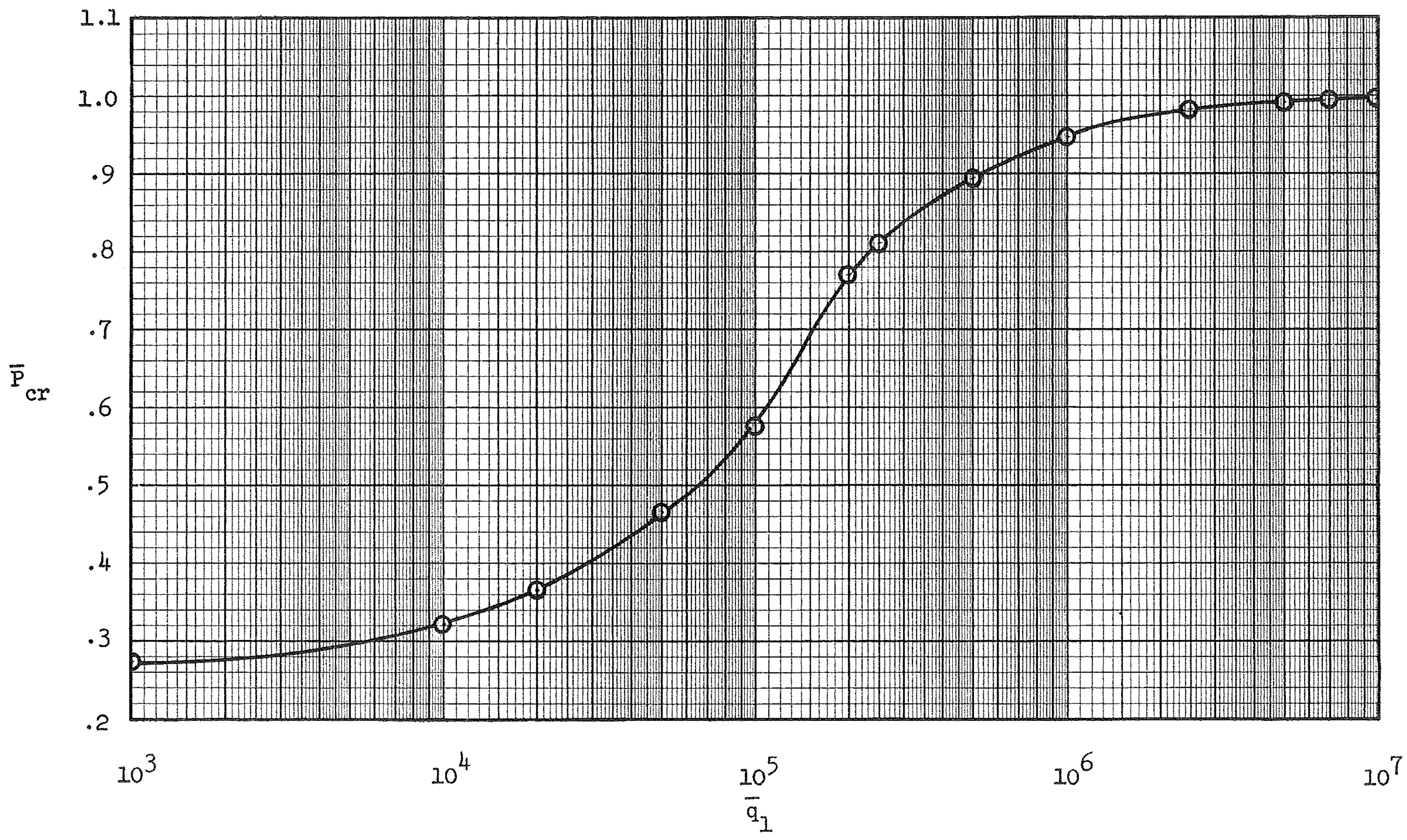

Figure 7.- Buckling load $\bar{P}_{cr}$ as a function of inplane restraint $\bar{q}_1$ for shell parameter $\mu = 6$, $\nu^2 = 0.1$, $b/t = 100$, $\bar{q}_2 = 0$, $\bar{q}_3 = 10^5$, $\bar{q}_4 = 10^5$.

Between these extremes the buckling load increases monotonically with an increase in in-plane restraint.

The fact that the buckling load increases with increasing meridional restraint as indicated by figures 5, 6, and 7 is not unexpected. The buckling phenomenon is preceded by a build-up of the membrane forces. For shallow shells the magnitude of the membrane forces is greatly influenced by the degree of meridional restraint.

It is of interest to note the size of a circular ring at the edge which would be required to approximate a rigid restraint to meridional displacement. By assuming that $\bar{q}_1 = 10^7$, $\nu^2 = 0.1$, $b/t = 100$, and that the ring and shell are made from the same material it is found that the ring radius required is approximately 50t, where t is the thickness of the shell.

Effect of rotational edge restraint.- Calculations were made to determine the influence of rotational restraint on the buckling behavior of shallow spherical shells. Plots of the buckling load $\bar{P}_{cr}$ for various values of the rotational restraint parameter $\bar{q}_4$ are shown in figures 8, 9, and 10 for geometric parameters $\mu = 4$, 5, and 6, respectively. The calculations were carried out for a shallow spherical cap with the meridional edge displacement fully restrained $\left(\bar{q}_1 = 10^7\right)$, no cross-coupling between normal and meridional edge displacements $\left(\bar{q}_2 = 0\right)$ and with $\nu^2 = 0.1$, $b/t = 100$. The plots show that the buckling load is essentially independent of the rotational restraint for $\bar{q}_4 < 0.01$ and

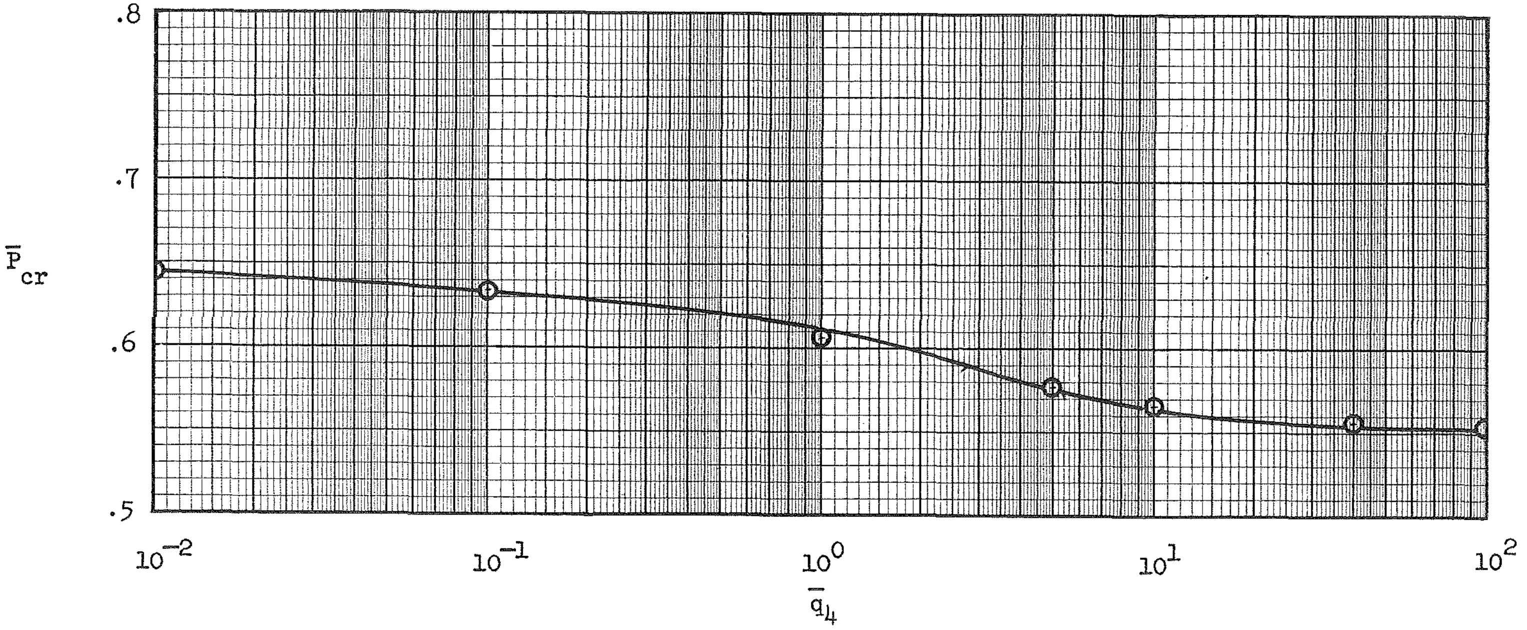

Figure 8.- Buckling load $\bar{P}_{cr}$ as a function of rotational restraint $\bar{q}_4$ for shell parameter $\mu = 4$, $\nu^2 = 0.1$, $b/t = 100$, $\bar{q}_1 = 10^7$, $\bar{q}_2 = 0$, $\bar{q}_3 = 10^5$.

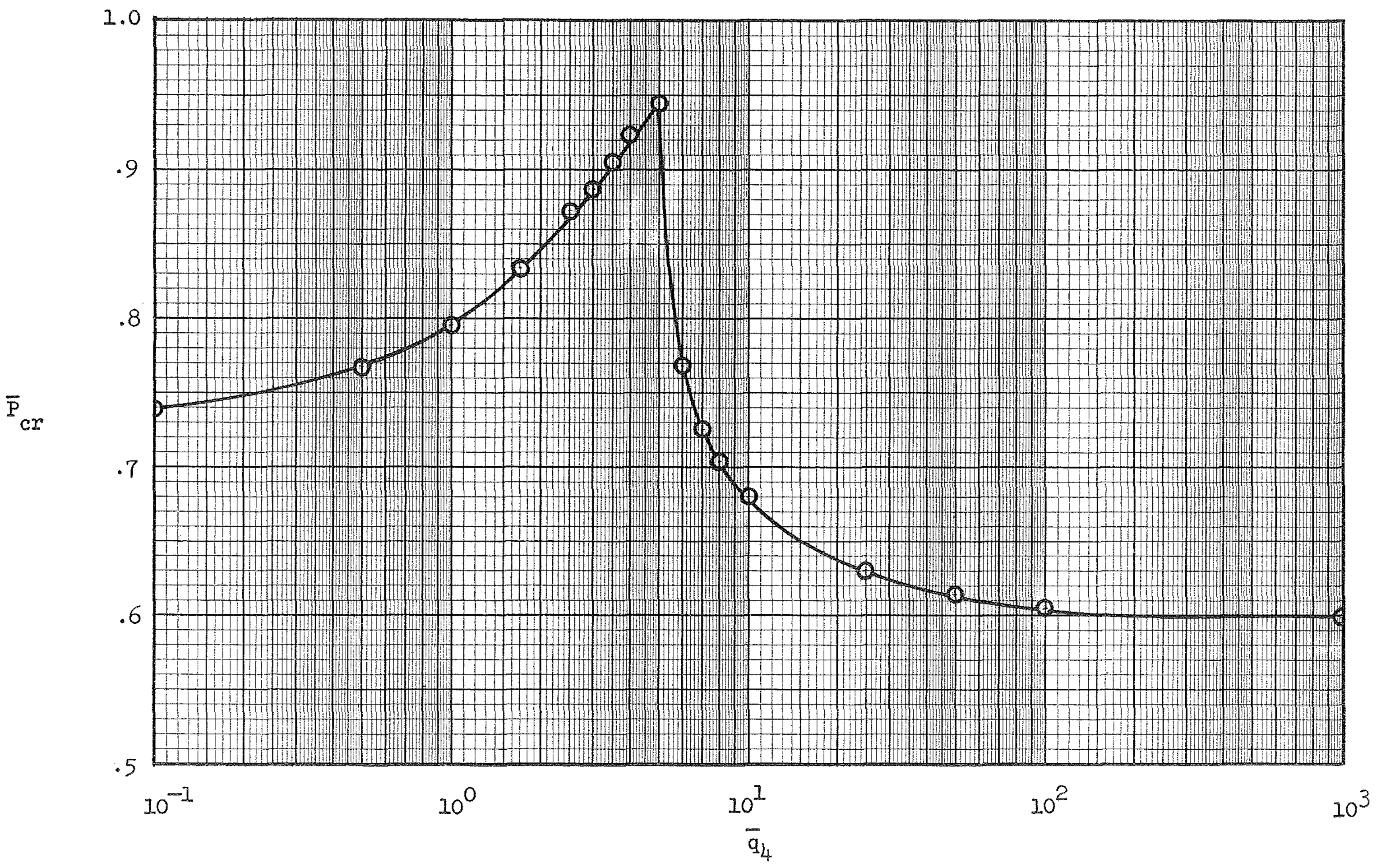

Figure 9.- Buckling load $\bar{P}_{cr}$ as a function of rotational restraint $\bar{q}_4$ for shell parameter $\mu = 5$, $\nu^2 = 0.1$, $b/t = 100$, $\bar{q}_1 = 10^7$, $\bar{q}_2 = 0$, $\bar{q}_3 = 10^5$.

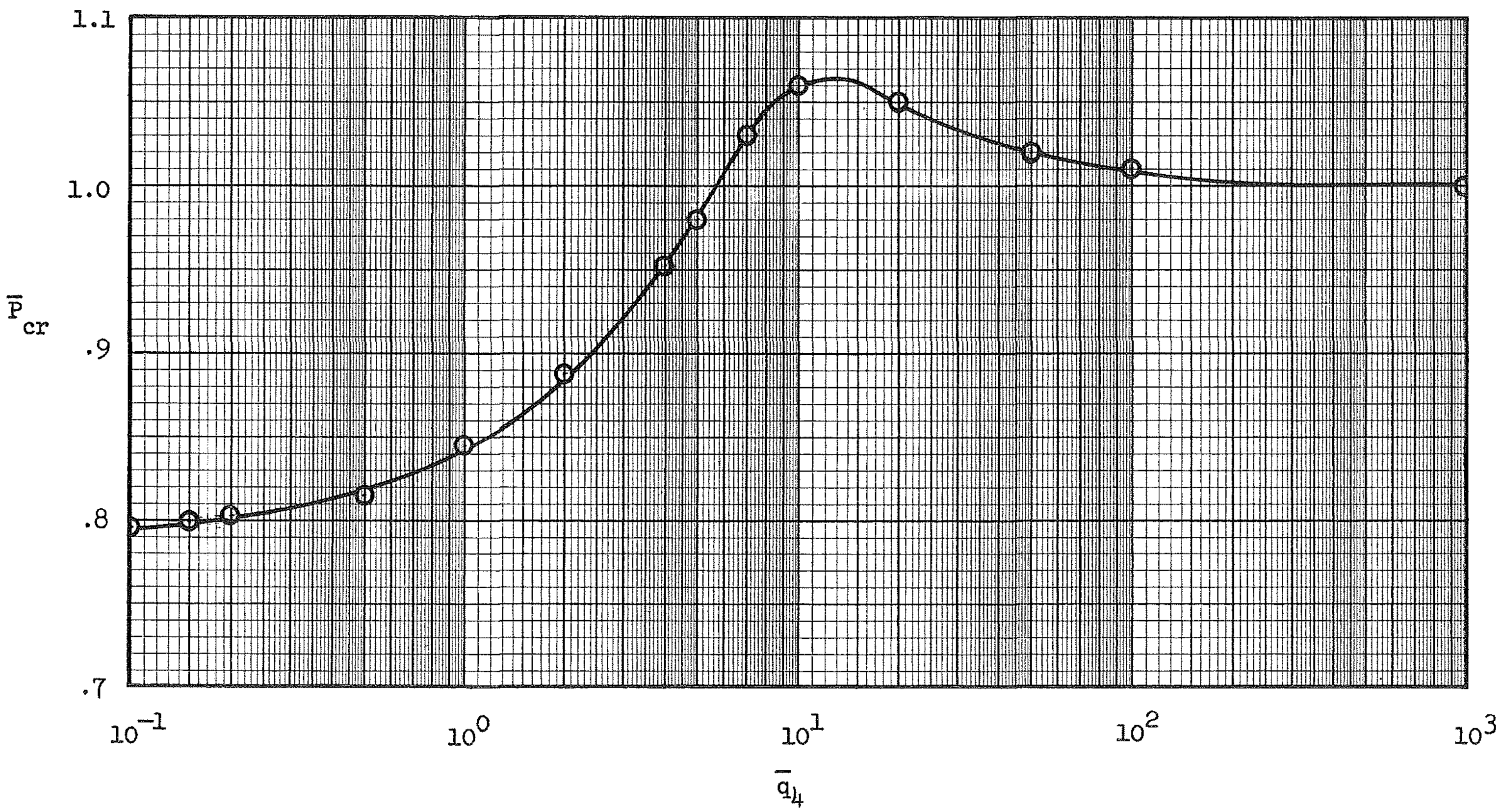

Figure 10.- Buckling load $\bar{P}_{cr}$ as a function of rotational restraint $\bar{q}_4$ for shell parameter $\mu = 6$, $\nu^2 = 0.1$, $b/t = 100$, $\bar{q}_1 = 10^7$, $\bar{q}_2 = 0$, $\bar{q}_3 = 10^5$.

$\bar{q}_4 > 10^3$. Between these extremes the character of the variation of buckling load with a change in the rotational restraint parameter is dependent on the value of the geometric parameter μ.

Figures 8, 9, 10 show two phenomena which are of interest. First, that for $\mu = 5$ and 6 a peak in the $\bar{P}_{cr}$ curve occurs between the extremes of clamped $\left(\bar{q}_4 > 10^3\right)$ and simply supported $\left(\bar{q}_4 < 0.01\right)$ edge conditions. Second, that for $\mu = 4$ and 5 the buckling load for the simply supported edge condition is greater then the buckling load for the clamped edge condition. While this behavior is unexpected it is believed that both effects can be explained to some degree by considering the normal deflection mode shapes. The mode shapes just prior to buckling were calculated for several combinations of the parameters. The results are shown in figures 11, 12, and 13 for geometric parameters $\mu = 4, 5$, and 6, respectively. The buckling modes for values of the rotational restraint parameter associated with the simply supported and clamped edge conditions are presented for $\mu = 4, 5$, and 6. In addition, the modes associated with values of the rotational restraint parameter just prior to and just after the peak in $\bar{P}_{cr}$ are presented for $\mu = 5$ and 6.

It is seen that the character of the simply supported buckling mode changes between $\mu = 4$ and $\mu = 5$ and that the character of the clamped buckling mode changes between $\mu = 5$ and $\mu = 6$.

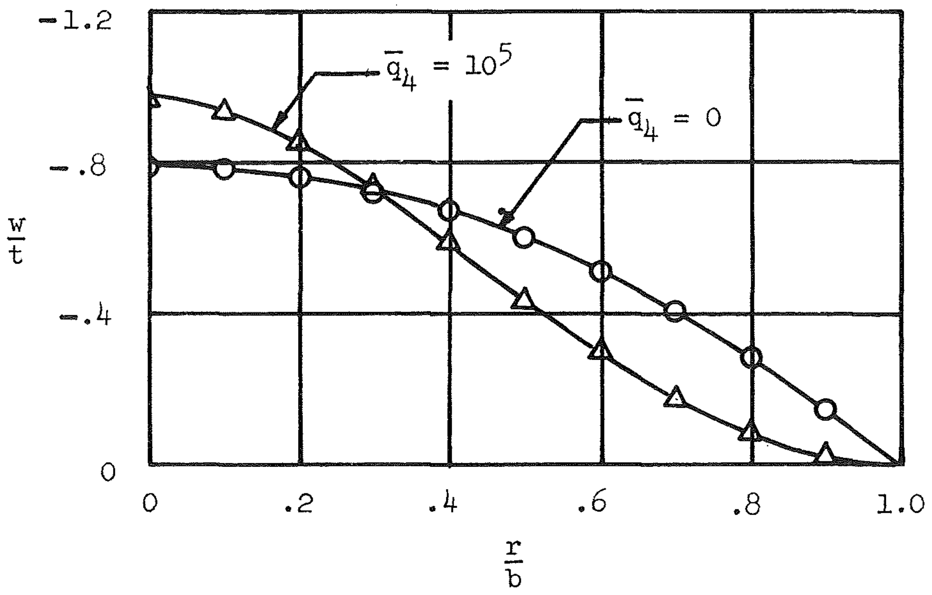

Figure 11.- Normal deflection as a function of radius for various values of rotational restraint $\bar{q}_4$ for $\mu = 4$, $\nu^2 = 0.1$, $\frac{b}{t} = 100$, $\bar{q}_1 = 10^7$, $\bar{q}_2 = 0$, $\bar{q}_3 = 10^5$.

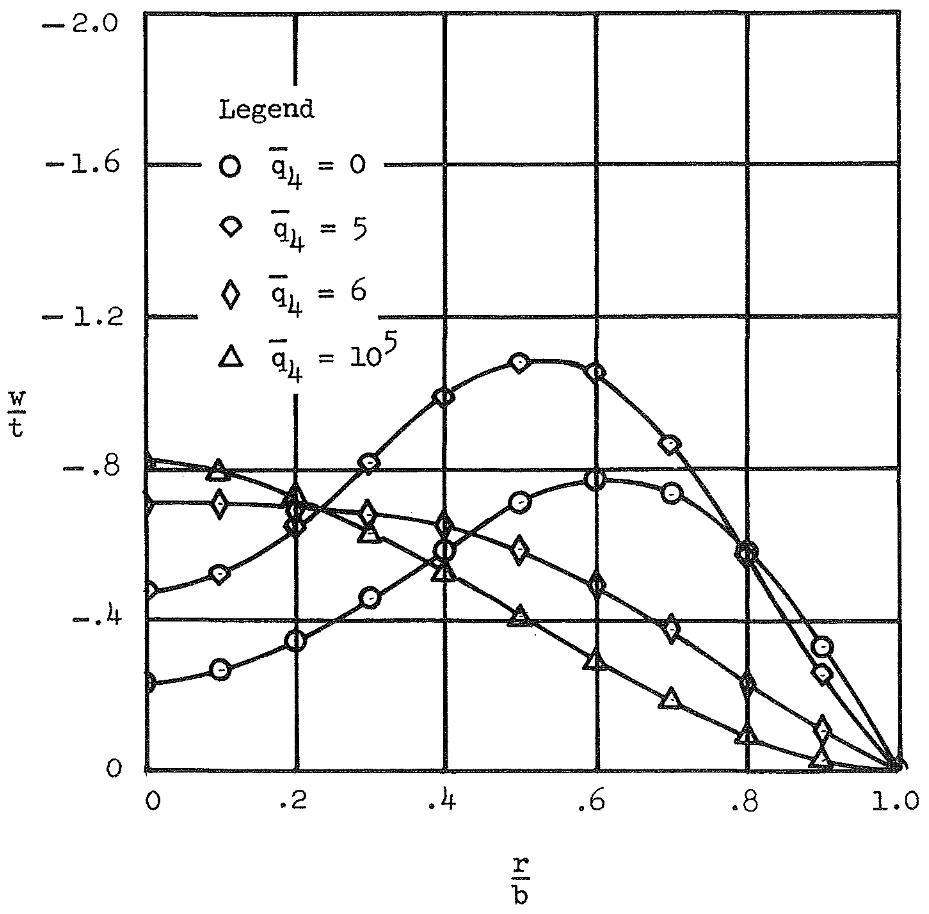

Figure 12.- Normal deflection as a function of radius for various values of rotational restraint $\bar{q}_4$ for $\mu = 5$, $\nu^2 = 0.1$, $b/t = 100$, $\bar{q}_1 = 10^7$, $\bar{q}_2 = 0$, $\bar{q}_3 = 10^5$.

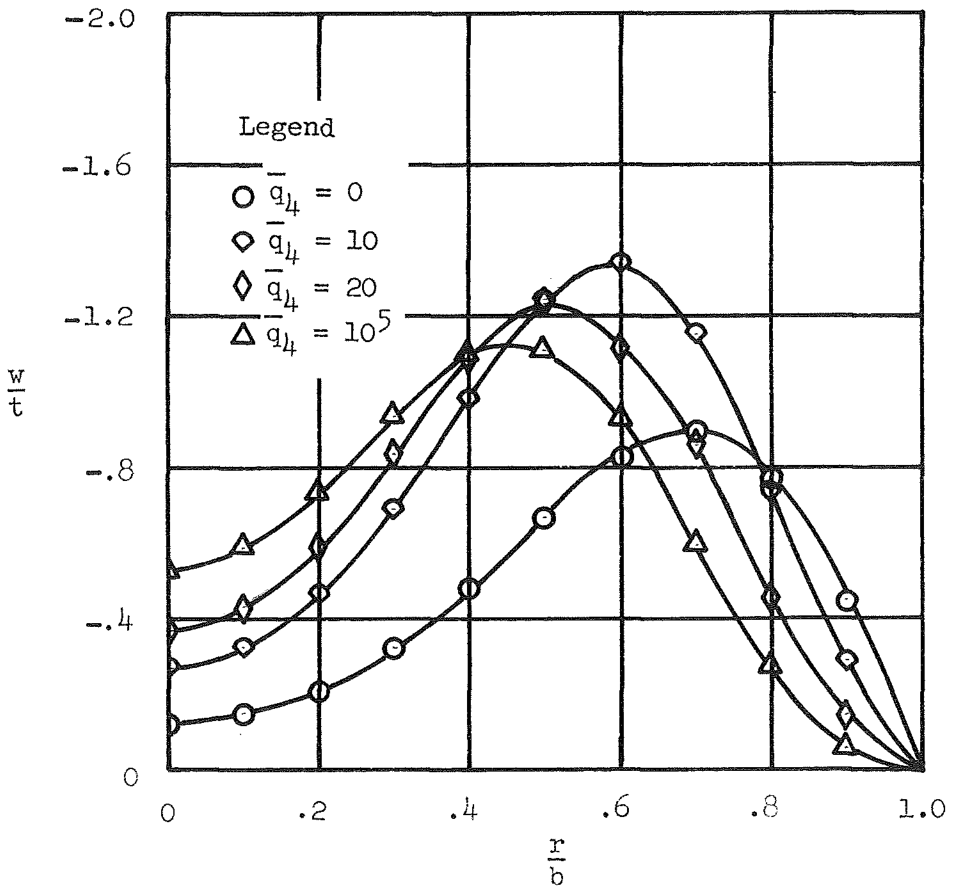

Figure 13.- Normal deflection as a function of radius for various values of rotational restraint $\bar{q}_4$ for $\mu = 6$, $\nu^2 = 0.1$, $\frac{b}{t} = 100$, $\bar{q}_1 = 10^7$, $\bar{q}_2 = 0$, $\bar{q}_3 = 10^5$.

In both cases the buckling mode changes from a mode characterized by one wave to a mode characterized by two waves. These changes in mode shape can be correlated with abrupt changes in the slope of the plots of buckling load versus shell parameter which are presented in figure 14 for simply supported and clamped edges. The calculations which are presented in figure 14 were carried out for shallow spherical shells with fully restrained meridional edge displacements $\left(\bar{q}_1 = 10^7\right)$, no cross-coupling between u_b and w_b $\left(\bar{q}_2 = 0\right)$ and with $\nu^2 = 0.1$, $b/t = 100$. Also $\bar{q}_4 = 10^5$ was taken to approximate completely restrained rotation, and $\bar{q}_4 = 0$ for the simply supported edge. Abrupt changes in slopes of the buckling load curves occur at $\mu = 4.2$ and $\mu = 5.8$ for the clamped and simply supported edges, respectively.

The geometric parameters at which the changes in slope of the buckling load curves occur for the clamped and the simply supported edge can be correlated by considering the clamped buckling mode. Consider for example the buckling mode for the clamped edge with $\mu = 6$ (fig. 12). A point of inflection occurs at a radius of approximately $r = 0.7b$. Since w'' is zero at the inflection point it follows that the moment also vanishes. This point therefore represents the edge of a shell which can be considered as having approximately simply supported edges although these conditions do not exactly correspond to those used in calculating the simply supported results presented in figure 14. Since μ is proportional to the outer radius it follows that the clamped shell

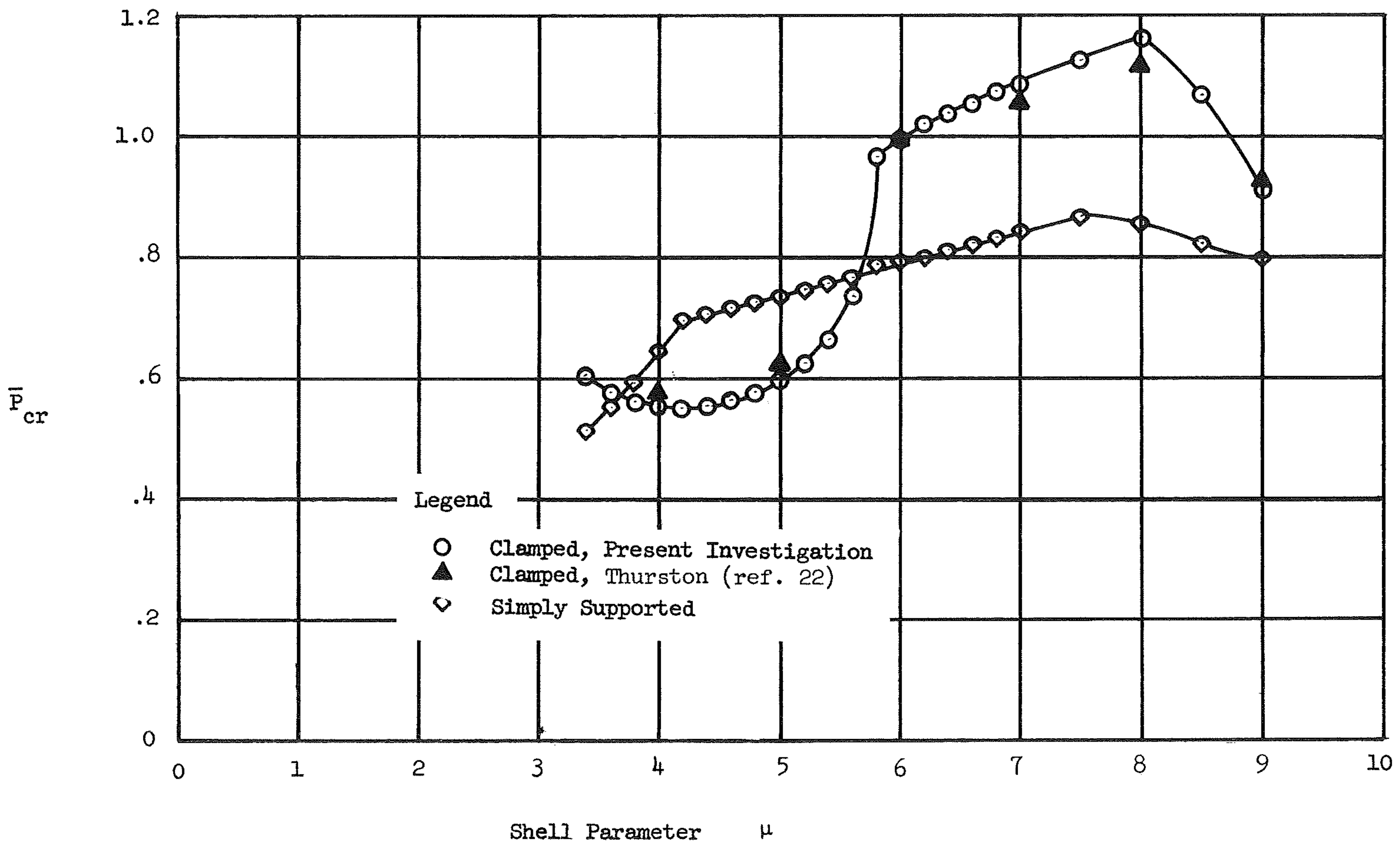

Figure 14.- Critical buckling load $\bar{P}_{cr}$ as a function of shell parameter μ for a shallow spherical shell with clamped and simply supported edges.

with $\mu = 6$ has some characteristics in common with a simple supported shell with $\mu = 4.2$. In fact, the mode shape for the clamped edge for $5.8 < \mu < 8$ corresponds to the mode shape for the simply supported shell for $4.2 < \mu < 7.5$. The buckling load is independent of b so that if the boundary conditions at the inflection point were the same as those used in calculating the results shown in figure 14 for the simply supported shell, the corresponding buckling loads would be the same.

The fact that the buckling load for the simply supported shell is greater than the buckling load for a clamped shell with the same μ is explainable since the buckling load for the simply supported shell can be correlated with the buckling load of an equivalent clamped shell of higher μ. Other investigators have noted that the spherical shell with a simply supported edge has a higher buckling load than the same shell with a clamped edge (see Evansen and Fulton (ref. 45), for example). The increase in buckling load was attributed to dynamic effects by these authors whereas the results of the present investigation show that the increase is due to the shell configuration.

The explanation for the peak in the variation in buckling load with rotational restraint is not quite so apparent. It does appear, however, that the results are not unreasonable in view of the variation of buckling load with μ shown in figure 14 and the fact that for any value of $\bar{q}_4$ there is an equivalent simply supported shell. The very sharp peak shown in figure 9 for $\mu = 5$ is evidently related to the fact that the mode shape for the simply supported shell associated with the clamped edge has one wave whereas the mode shape for the simply supported edge has two waves.

It is of interest to note the size of a circular ring at the edge which would be required to approximate a rigid rotational restraint $\left(\bar{q}_4 > 10^3\right)$. A ring radius of approximately 10t is found adequate for $\bar{q}_4 = 10^3$, $\nu^2 = 0.1$, b/t = 100. It is also assumed that the ring and shell are made from the same material and that the shell radius b is 10 times the radius of the ring. It was found previously that a ring radius of approximately 50t is required to simulate a rigid restraint to meridional displacement. It would thus appear that in designing a supporting ring to simulate clamped edge conditions the meridional restraint is the controlling factor.

Correlation of results with other investigators.- The variation of the buckling load with shell parameter for a clamped edge can be compared to results which have been presented by other investigators (see Thurston, ref. 22, for example). The buckling loads as determined by the present investigation are seen to be in good agreement with the buckling loads presented by Thurston, which are indicated by the solid symbols in figure 14. The slight differences in the buckling loads predicted by the present investigation and by Thurston may be due to the limited number of increments used in the present investigation (10) compared to the number used by Thurston (18). Another possible cause is the different discretization error in each analysis. The present analysis evaluates first derivatives midway between stations while Thurston evaluates these derivatives at the station. The generally good agreement between the buckling loads as determined by the direct and the indirect methods of analysis indicates that the direct method as presented in this investigation is a valid and accurate numerical technique for solving problems in the field of continuum mechanics.

Elliptic shallow shell.- The effects of changes in shallow shell geometry on the buckling load are determined by considering a shallow elliptic shell of revolution. The shallow elliptical shell is formed by revolving the curve

$$\left(\frac{r}{c}\right)^2 + \left(\frac{z}{a}\right)^2 = 1 \tag{144}$$

about the z-axis and taking that part of the surface $0 \leq r \leq b$. The coordinate system is as shown in figure 15.

Equation (144) can be written in the following form

$$\rho^2 + \eta^2\zeta^2 = \beta^2 \tag{145}$$

where

$$\eta = \frac{c}{a}$$

$$\beta = \frac{c}{b}$$

$$\rho = \frac{r}{b}$$

$$\xi = \frac{z}{b}$$

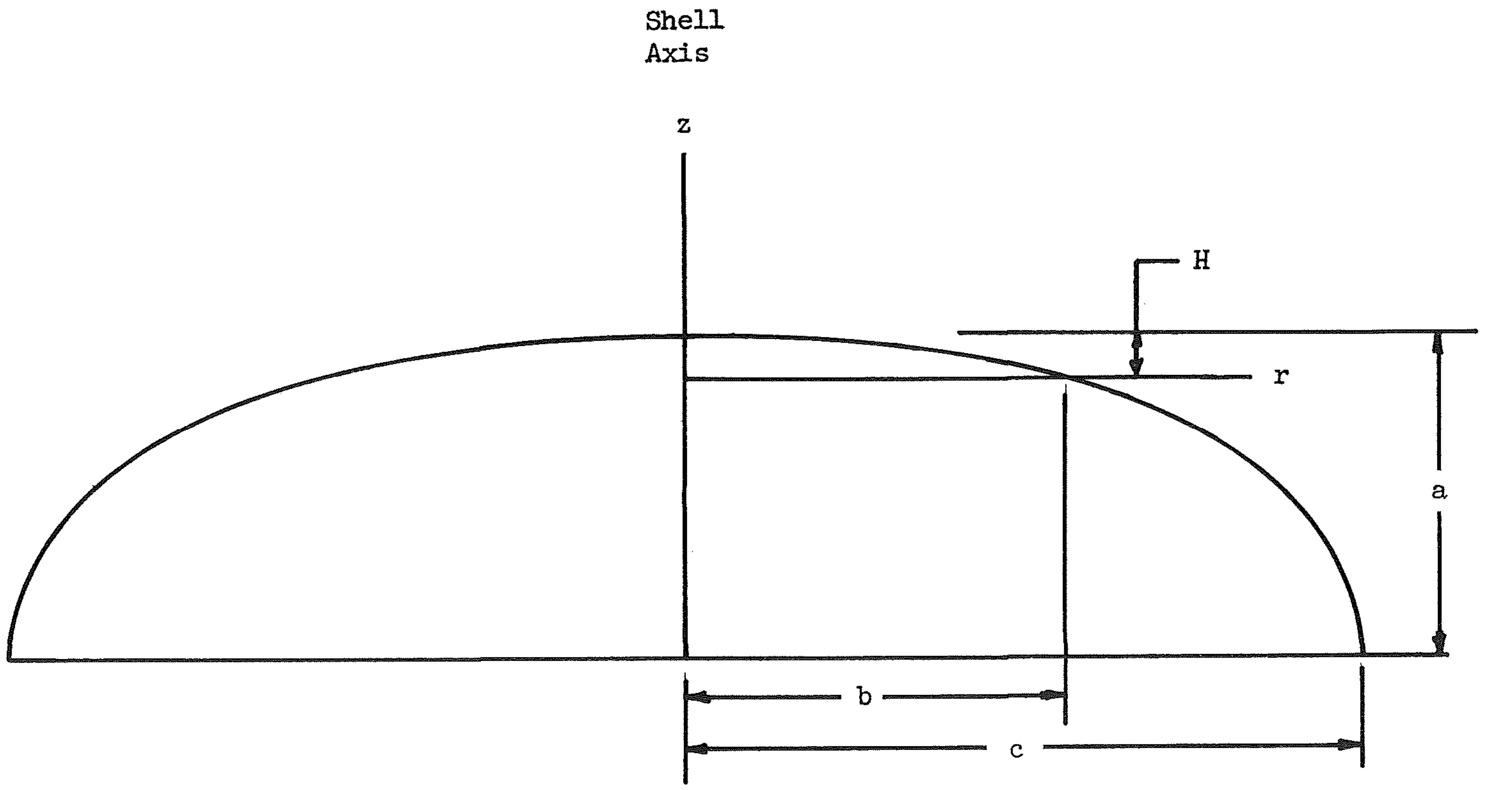

Figure 15.- Shallow elliptical shell geometry.

TABLE I

EFFECT OF CHANGE IN SHELL GEOMETRY ON BUCKLING LOAD AND CENTER DEFLECTION FOR A UNIFORMLY LOADED CLAMPED SHALLOW SHELL WITH $b/t = 100$, $\nu^2 = 0.1$

Spherical shell parameter	η	$\frac{w_o}{t}$	$\overline{P}_{cr}$
4	0.5	- 0.9745	0.5544
4	1	- .9271	.5528
4	2	- .9283	.5544
5	.5	- .8198	.5976
5	1	- .8274	.5976
5	2	- .8348	.5992
6	.5	- .5279	.9984
6	1	- .5252	.9952
6	2	- .5542	1.0112

The principal curvatures ω_1, ω_2 are given by the following

$$\omega_1 = \frac{1}{\eta^2 \zeta} \left(1 + \frac{1}{\eta^2} \frac{\rho^2}{\zeta^2} \right) \left\{ 1 + \frac{1}{\eta^4} \frac{\rho^2}{\zeta^2} \right\}^{-\frac{3}{2}} \tag{146}$$

$$\omega_2 = \frac{1}{\eta^2 \zeta} \left\{ 1 + \frac{1}{\eta^4} \frac{\rho^2}{\zeta^2} \right\}^{-\frac{1}{2}} \tag{147}$$

At $\rho = 0$ the curvatures are taken to be equal to ω_o. It follows that ζ may be expressed in terms of η and ω_o as follows

$$\zeta = \frac{1}{\eta^2 \omega_o} \left\{ 1 - \eta^2 \omega_o{}^2 \rho^2 \right\}^{\frac{1}{2}} \tag{148}$$

The effect of changes from a nominal spherical shell shape on the buckling load is found by specifying ω_o as the curvature for a spherical cap and varying η. The center deflection $\frac{w_o}{t}$ and the buckling load $\overline{P}_{cr}$ for various elliptical caps with nominal spherical geometric parameters $\mu = 4, 5,$ and 6 are given in table I.

The results presented in table I indicate that for the range of parameters considered there is essentially no change in the center deflection or the buckling load for a given value of the spherical shell parameter μ when η is varied between 0.5 and 2. The maximum change of center deflection is approximately 5 percent for $\mu = 4$. The maximum change in the buckling load is less than one-half of 1 percent.

XI. CONCLUDING REMARKS

Mathematical programing techniques for minimizing functions of many variables are evaluated in order to determine which methods are appropriate in the field of continuum mechanics. The methods which are found to be most appropriate are applied to determine the nonlinear displacements of shallow shells of revolution.

The mathematical programing techniques considered in the investigation were the steepest descent, the conjugate gradient, the variable metric, and the Newton-Raphson method. Of the methods considered it was found that a combination of the Newton-Raphson and the conjugate gradient methods was best suited to the numerical solution of nonlinear problems in continuum mechanics. The Newton-Raphson procedure is generally used to find a minimum of the object function. The conjugate gradient method was used to determine an initial approximation sufficiently close to the location of the minimum such that the Newton-Raphson procedure would converge.

The combined algorithm was applied to determine the large deflection behavior of a symmetric shallow shell loaded uniformly and having arbitrary edge restraints. The approach used in this investigation was to obtain the displacements by a direct minimization of the finite-difference approximation of the potential energy of the shell. The resulting displacement set defines an equilibrium configuration of the shell as a consequence of the Theorem of Minimum Potential. The set of equations which defines the gradient has the advantage that the elements are symmetric and also may be more accurate then the finite-difference approximations of the equilibrium equations.

The effect of boundary restraint on the buckling of a shallow spherical shell was investigated. It was found that in the range $3.5 < \mu < 5.6$ the simply supported shell has a higher buckling load than a clamped shell with the same value of μ. In addition, for $\mu > 5$ it was found that partial restraint of edge rotation can lead to higher buckling loads than results from no restraint (simply supported) or fully restrained (clamped).

The effect of changes in shallow shell geometry on the buckling load was investigated by considering a portion of a shallow ellipse of revolution. For the range of parameters considered, it was found that the buckling load and center deflection were essentially unchanged when the ratio of ellipse major radii was varied from 0.5 to 2. For a given value of μ the maximum change in the buckling load was less then one-half of 1 percent and the maximum change in center deflection was approximately 5 percent.

XII. ACKNOWLEDGEMENTS

The author wishes to express his thanks and appreciation to the National Aeronautics and Space Administration for the educational opportunities which have been provided and for permitting him to perform this research as a part of his work assignment at the Langley Research Center. He also wishes to thank Dr. D. Frederick, Professor of Engineering Mechanics, Virginia Polytechnic Institute, and Dr. R. A. Fulton of the N.A.S.A. for their guidance and councel. Finally he wishes to express deep gratitude and thanks to his wife for the patience and understanding that she has shown, and moral encouragement which she has provided.

XIII. REFERENCES

1. Rayleigh, Lord: Theory of Sound. Dover, New York, 1945.

2. Timoshenko, S.; and Woinowsky-Krieger, S.: Theory of Plates and Shells, McGraw-Hill, New York, 1959.

3. Ritz, W.: Ueber eine Methode zur Lösung gewisser Variation problems der mathematischen Physik. Jour. reine angew. Math., 135, 1909, pp. 1-61.

4. Timoshenko, S.; and Young, D. H.: Vibration Proglems in Engineering. D. Von Nostrand Co., New York, 1955.

5. Ketter, D. J.: Flutter of Flat, Rectangular, Orthotropic Panels. AIAA Jour., Vol. 5, No. 1, Jan. 1967, pp. 116-124.

6. Huang, T. C.: Application of Variational Methods to the Vibration of Plates Including Rotary Inertia and Shear. In Developments in Mechanics, Proceedings of Seventh Midwestern Mechanics Conference, Sept. 6-8, 1961, Michigan State Univ., Ed. by Lay, J.E., and Malvun, L.E., Plenum Press, New York, 1961, pp. 61-72.

7. Salvadori, M. G.; and Baron, M. L.: Numerical Methods in Engineering. Prentice Hall, Englewood Cliffs, N. J., 1961, pp. 190-293.

8. Houbolt, John C.: A Study of Several Aerothermoelastic Problems of Aircraft Structures in High Speed Flight. Nr. 5, Mitteilungen aus dem Institut für Flugzeugstatik und Leichtbau. Leeman (Zürich) c. 1958 p. 47-58.

9. Walton, William C., Jr.: Application of a General Finite-Difference Method for Calculating Bending Deformations of Solid Plates. NASA TN D-536, 1960.

10. Schaeffer, H. G.; and Heard, W. L., Jr.: Evaluation of an Energy Method Using Finite Differences for Determining Thermal Midplane Stresses in Plates. NASA TN D-2439, 1964.

11. Bogner, F. K.; Mallett, R. H.;Munich, M. D.; and Schmidt, L. A.: Development and Evaluation of Energy Search Methods of Nonlinear Structural Analysis. AFFDL-TR-65-113, October, 1965.

12. Reissner, H.: Spannungen in Kugelschlen (Kuppeln). Müller-Breslau-Festschrift, 1912, p. 181.

13. Marguerre, K.: Zur Theore der Gekrummten Platte Grosser Formanderung. Proc. Fifth International Congress of Applied Mech., 1938, pp. 93-101.

14. Reissner, E.: Stresses and Small Displacements of Shallow Spherical Shells. (I). Jour. Math. and Phys., Vol. 25, 1946, pp. 80-85, (II). Jour. Math. and Phys., Vol. 25, 1947, pp. 279-300.

15. Kaplan, A.; and Fung, Y. C.: A Nonlinear Theory of Bending and Buckling of Thin Elastic Shallow Spherical Shells. NACA TN 3212, August 1954.

16. Archer, R. R.: Stability Limits for a Clamped Spherical Shell Segment Under Uniform Pressure. Quart. Appl. Math., Vol. 15, January 1958, p. 335.

17. Reiss, E. L.; Greenburg, H. J.; and Keller, H. B.: Nonlinear Deflections of Shallow Spherical Shells. Jour. Aero. Sciences, Vol. 54 July 1957, 1953.

18. Weinitschke, H. J.: On the Nonlinear Theory of Shallow Spherical Shells. Jour. Soc. Indust. and Applied Math., Vol. 6, September 1958, p. 209.

19. Chen, W. L.: Effect of Geometrical Imperfection of the Elastic Buckling of Snallow Spherical Shells. Sc. D. Thesis, Dept. of Civil and Sanitary Engineering, Mass. Inst. of Tech. Jan. 1959.

20. Von Wellick, G. P. R.: The Elastic Stability of Thin Spherical Shells. Jour. Engr. Mech. Div., Proc. A.S.C.E., Vol. 185, No. EM1, Jan. 1959.

21. Budiansky, B.: Buckling of Clamped Shallow Spherical Shells. Proceedings of the I.U.T.A.M. Symposium of the Theory of Thin Elastic Shells (North Holland Publishing Co., Amsterdam, 1960) pp. 64-94.

22. Thurston, G. A.: A Numerical Solution of the Nonlinear Equations for Axisymmetric Bending of Shallow Spherical Shells. Jour. Applied Mech., Dec. 1961, pp. 557-562.

23. Archer, R. R.: On the Numerical Solution of the Nonlinear Equations for Shells of Revolution. Jour. Math. and Phys. Vol. 41, 1962, pp. 165-178.

24. Mescall, J. F.: Numerical Solution of the Nonlinear Axisymmetric Equations for Shells of Revolution. AMRA TR 66-11, May 1966.

25. Wang, L. R. L.; Rodriquez-Agrait, L.; and Litle, W. A.: Effect of Boundary Conditions on Shell Buckling. Jour. Engr. Mech. Div. of Proc. of ASCE, Vol. EM6, Dec. 1966, pp. 101-116.

26. Spang, H. A.: A Review of Minimization Techniques for Nonlinear Functions. The Soc. of Indust. and Appl. Math. Review, Vol. 4, No. 4, Oct. 1962, pp. 343-365.

27. Cauchy, A.: Methode Generale Pour La Resolution des Systems d'Equations Simultaneous. Comp. Rand, L'Academic Sciences, 25, 1847, p. 536.

28. Rosenbrock, H. H.: An Automatic Method for Finding the Greatest or Least Value of a Function. The Computer Jour. Vol. 3, 1960, pp. 175-184.

29. Hestenes, M. R.; and Stiefel, E.: Methods of Conjugate Gradients for Solving Linear Systems. Jour. Res. Nat. Bureau of Standards, Vol. 49, No. 6, Dec. 1952, pp. 409-436.

30. Beckman, F. S.: The Solution of Linear Equations by the Conjugate Gradient Method. Mathematical Methods for Digital Computers edited by Ralstan, A.; and Welf, H. S.; Wiley, 1960, pp. 62-72.

31. Fletcher, R.; and Reeves, C. M.: Function Minimization by Conjugate Gradients. Computer Journal, Vol. 7, No. 2, July 1964, pp. 149-154.

32. Powell, M. J. D.: An Iterative Method for Finding Stationary Values of a Function of Several Variables. Computer Journal, Vol. 5, 1962, pp. 147-151.

33. Davidon, W. C.: Variable Metric Method of Minimization. Argonne Natl. Lab. Report ANL-5990, May 1959.

34. Fletcher, R.; and Powell, M. J. D.: A Rapidly Convergent Descent Method for Minimization. Computer Journal, Vol. 6, 1963, 1964, pp. 163-168.

35. Tomkins, C. B.: Methods of Steep Descent. Chapt. 18, Modern Mathematics for the Engineer, edited by Beckenback, E. F.; McGraw Hill, 1956, pp. 448-479.

36. Frederick, D.; and Chang, T. S.: Continuum Mechanics. Allyn and Bacon, Boston, 1965, pp. 7-16.

37. Bryson, A. E.; and Denham, W. F.: A Steepest Ascent Method for Solving Mathematical Programing Problems. Jour. Appl. Mech., June 1962, pp. 247-258.

38. Kelly, H. J.: Gradient Theory of Optimal Flight Paths. American Rocket Society Journal, Oct. 1960, pp. 947-954.

39. McGill, R.; and Kenneth, P.: Solution of Variational Problems by Means of a Generalized Newton-Raphson Operator. AIAA Jour., Vol. 2, No. 10, Oct. 1964, pp. 1761-1766.

40. Langhaar, H. L.: Energy Methods in Applied Mechanics. Wiley, 1962, p. 161.

41. Mushtari, Kh., M.; and Galemov, K. Z.: Nonlinear Theory of Thin Elastic Shells. NASA TT-F62, 1961.

42. Cyrus, N. J.; and Fulton, R. E.: Finite Difference Accuracy in Structural Analysis. Jour. of the Struct. Div., Proc. ASCE, Vol. 92, No. ST6, Dec. 1966, pp. 459-471.

43. White, R. N.: Optimal Solution Techniques for Finite Difference Equations. Jour. Struct. Div., Proc. A.S.C.E., August 1963, pp. 115-136.

44. Huang, N. C.: Unsymmetrical Buckling of Shallow Spherical Shells. AIAA Jour. Vol. 1, No. 4, April 1963, p. 945.

45. Evensen, D. A.; and Fulton, R. E.: Some Studies on the Nonlinear Dynamic Response of Shell-Type Structures. Dynamic Stability of Structures, Proceedings of International Conference, Evanston, Illinois, Oct. 18-20, 1965, Pergamon Press, 1966, pp. 287-253.

XIV. VITA

Harry G. Schaeffer was born in Clear Lake, Washington, on June 2, 1934. He attended high school in Bremerton, Washington, and graduated in 1952. He entered the University of Washington upon graduation from high school and graduated with a Bachelor of Science degree in Mechanical Engineering in June 1958.

He then accepted a position with the AiResearch Manufacturing Company of Arizona and moved to Phoenix, Arizona. In the fall of 1958 he entered the Arizona State University at Tempe, Arizona, and pursued graduate work in Engineering Science on a part-time basis. He graduated from Arizona State in 1962 with a Master of Science degree in Engineering Science.

After completing the requirements for the master's degree he accepted a position with the Langley Research Center of the National Aeronautics and Space Administration and moved to Hampton, Virginia. His present position at the Langley Research Center is that of an Aerospace Engineer in the Structures Research Division.

Harry G. Schaeffer

XV. APPENDIX

COMPUTER PROGRAM FOR DIRECT DETERMINATION OF NONLINEAR DISPLACEMENTS AND BUCKLING LOADS FOR AXISYMMETRICALLY LOADED SHELLS OF REVOLUTION

A computer program which minimizes the finite-difference approximation of the shell potential energy (eq. (116)) is described. This program is written in FORTRAN IV symbolic language for operation on the IBSYS-IBJOB operating system (version 13).

Input Data

The following input data cards are required for each separate problem.

Card number 1 The description of problem, columns 1-80.

Card number 2 Card number 2 is a card or group of cards necessary to read a group called NAM1, using the NAMELIST input option. These data are as follows.

ANU	Poissons ratio
SF	Scale factor, l
ALD(1)	Initial load, $\bar{P}$
ALD(2)	Load increment, $\Delta\bar{P}$
ALD(3)	Maximum load
BBC(1)	Membrane stiffness, k

DDC(1) Bending stiffness, d

OXC(2) Ratio of principal radii of ellipse, η

NO Problem number

NINC Number of increments along radius (maximum of 10)

DELTA Convergence criterion for angle between $g_i^{\alpha+1}$ and p_i^{α} in one-dimensional search (see eq. (75)

NNL The number of cycles taken in the conjugate gradient search before checking to determine if the Newton-Raphson procedure converges is NNL times n where n is the number of variables. Recommended number is NNL = 3.

ERR Conjugate gradient convergence criterion Recommended value ERR = 5×10^{-7}

EST Estimated value of minimum of function

IPF Controls print frequency in conjugate gradient search. Recommended value, IPF = n.

AMU Spherical shell parameter μ

BOH b/t

Q(1) $\bar{q}_1$

Q(2) $\bar{q}_2$

Q(3) $\bar{q}_3$

Q(4) $\bar{q}_4$

PN(1) Load parameter set PN(1) = - 1

IXIF1 Normal setting is zero. If IXIF1 = 1 an initial displacement state is to be read in on the next set of data cards

NREAD Normal setting is zero. If NREAD > 0 the Newton-Raphson procedure is bypassed, an initial state vector is read and the conjugate gradient method is initiated

IPIF1 Intermediate print control. Normal setting IPIF1 = 0. For debugging purposes set IPIF1 = 1

Card number 3 Necessary only if IXIF1 = 1 or NREAD > 0. Initial displacement set XX(I) (I = 1, n) read in by NAMELIST file called NM2.

Program Output

The normal output consists of the following

(1) Program description and important parameters

(2) Linear solution

(3) Converged nonlinear solution at specified load increments. Results include value of the potential FSTAR, the norm of the gradient GNORM, the value of the load $\bar{P}$ PBAR, the displacement set XX and the gradient set GG. The first NINC+2 components of XX and GG are associated with w-displacements, and the last NINC components are associated with u-displacements.

(4) Intermediate data from conjugate gradient search if applicable. Includes cycle number, count, together with load, displacement and gradient sets, value of the potential and norm of gradient.

Addition program output will occur if IPIF1 = 1. This additional output is used mainly for code checking and includes the following quantities.

1. Linear matrix elements, equations (124)-(134)
2. Nonlinear matrix elements, equations (135)-(140)
3. Intermediate data from one-dimensional search

Description of Program

The program consists of a main calling sequence which controls the load incrementing and tests for convergence. In addition, the following subroutines are called during the execution of the program.

FUMICO	One-dimensional search
CYCLE	Conjugate gradient iteration
ITER	Initialize for Newton-Raphson procedure
OUTPUT	Prints potential, norm of gradient gradient and displacement
XVEC	Calculates $x^{\alpha+1}$
WOGR	Calculates $\frac{\partial\Pi}{\partial\omega_{i+1}}$, $\frac{\partial\Pi}{\partial\omega_{i-1}}$
UOGR	Calculates $\frac{\partial\Pi}{\partial u_{i+1}}$, $\frac{\partial\Pi}{\partial u_{i-1}}$

WOIGR	Calculates $\frac{\partial \bar{\Pi}}{\partial w_i}$
WOMPG	Calculates $\frac{\partial \bar{\Pi}}{\partial w_{i-2}}$, $\frac{\partial \bar{\Pi}}{\partial w_{i+2}}$
UOI	Calculates $\frac{\partial \bar{\Pi}}{\partial u_i}$
MOVEX	Change XX from a one-dimensional to a three-dimensional array
GRAD	Calculates gradient components
POTEN	Calculates potential energy
PILOAD	Calculates load potential
PIEDGE	Calculates restraint potential
LOADGR	Calculates gradient of load potential
EDGEGR	Calculates gradient of restraint potential
FDH	Calculates first derivative midway between stations
FDI	Calculates first derivative at the station
SD	Calculates second derivative
AVE	Calculates the displacement midway between stations
HGS	Calculates error vector in Newton-Raphson procedure
LMATC	Calculates matrix of second derivatives for Newton-Raphson procedure
AM, AP, AE, AMH, APH	Calculate the second derivatives of $\bar{\Pi}_L$

NMH, NPH	Calculate the second derivatives of $\bar{\Pi}_{NL}$
B	Calculate membrane stiffness
D	Calculate bending stiffness
O1	Calculate ω_1
O2	Calculate ω_2

Program Listing

The FORTRAN statements, which comprise the program as briefly, described are as follows.

```
      COMMON/BL1/PN(10)/BL2/Q(4)/BL3/OXC(3)/BL4/XX(633)/BL5/OTC(3)/BL6/
     1BBC(3)/BL8/GG(633)/BL9/DDC(3)           /BL17/ANU,SF,OMEGA/BL18/ALD
     2(4)/BLK1/XBAR(633)/BLK2/GBAR(633)/BK1/X(633)/COEF/AMAT(14)/NLCOEF/
     3ENMAT(9)/BL1/ML/BL15/W(21,10,3)/BL16/NINC,NFP1/BLK3/PIBAR
     4/BK2/IPLUS/BL7/P(633)           /BK3/IPIF1,NCL,DELTA/NONLIN/WN,UN,FN
      NAMELIST/NAM1/ANU,SF,ALD,OXC,OTC,BBC,DDC,NO,IXIF1,NINC,DELTA,IPIF1
     2,NNL,EST,ERR,NREAD,IPF,AMU,BOH
     1,Q,PN/NM2/XX
      DIMENSION ALDO(4),DDCO(3),XOLD(633)
      EQUIVALENCE (DOM,ALD(2)),(DOMO,ALDO(2))
  210 FORMAT(2X7H FSTAR=E16.8,2X3H F=E16.8)
      AK=0.
      KOUNT=0
      KNL=0
      NCL=2
      DELTQ=.04
      IPLUS=0
      DO 100 I=1,3
      OXC(I)=0.
      OTC(I)=0.
      BBC(I)=0.
  100 DDC(I)=0.
      DO 110 I=1,10
  110 PN(I)=0.D0
      PIBAR=0.
      DELTA=0.
      EST=0.
      ERR=0.
      IXIF1=0
      IPIF1=0
      NREAD=0
      NFP1=1
      ML=1
    2 READ(5,NAM1)
      READ(5,200)
  200 FORMAT(80H
     1                             )
      WRITE(6,201)
  201 FORMAT(1H1,47H NONLINEAR AXISYMMETRIC SHALLOW SHELL SCHAEFFER//)
      WRITE(6,200)
      ND=2*NINC+2
      OMZ=AMU**2/BOH/SQRT(12.*(1.-ANU**2))
      OXC(1)=OMZ
      AMU2=BOH*SQRT(12.*(1.-ANU**2))/AMU**4/4.
      WRITE(6,206) ANU,SF,DELTA,NINC,ND,AMU2
  206 FORMAT(1H0,5H ANU=E16.8,4H SF=E16.8,7H DELTA=E16.8,6H NINC=I5,4H N
     1D=I5,6H AMU2=E16.8)
      WRITE(6,207) ALD,OXC,OTC,BBC,DDC,Q,PN
  207 FORMAT(1H0,18H LOAD PARAMETERS  // 4E16.8 //13H XI CURVATURE// 3E1
     16.8 //16H THETA CURVATURE// 3E16.8 //19H MEMBRANE STIFFNESS// 3E16
     2.8 //18H BENDING STIFFNESS// 3E16.8 //20H BOUNDARY RESTRAINTS// 4E
     316.8 //18H LOAD COEFFICIENTS// 10E12.4 )
      DO 108 I=1,ND
      XBAR(I)=0.
  108 XX(I)=0.
      IF(IXIF1.EQ.1) READ(5,NM2)
      DO 101 I=1,3
      ALDO(I)=ALD(I)
  101 DDCO(I)=DDC(I)
      Q04=Q(4)
      IF(NREAD.EQ.1) GO TO 303
      NLIF1=0
       OMEGA=ALD(1)/AMU2
      CALL GRAD(FSTAR,0.,0.)
      CALL OUTPUT(ND,FSTAR)
    1 F=FSTAR
      CALL ITER(ND,XSTAR,XB)
      CALL GRAD(FSTAR,1.,0.)
      WRITE(6,210) FSTAR,F
      IF(NLIF1-1) 317,318,318
  317 WRITE(6,204)
  204 FORMAT(1H0,23H LINEAR BENDING RESULTS)
      CALL OUTPUT(ND,FSTAR)
      DO 105 I=1,ND
  105 XOLD(I)=XX(I)
      NLIF1=1
      GO TO 1
  318 IF(ABS((FSTAR-F)/FSTAR).LT.5.E-7)  GO TO 300
      IF(FSTAR-F)300,300,301
  301 IF(DOM-DELTQ*DOMO) 303,303,302
  302 OMBAR=ALD(1)-ALD(2)
      DOM=.2*DOM
      ALD(1)=OMBAR+DOM
       OMEGA=ALD(1) /AMU2
      DO 102 I=1,ND
  102 XX(I)=XOLD(I)
      CALL GRAD(FSTAR,1.,0.)
      CALL OUTPUT(ND,FSTAR)
      GO TO 1
  300 NLIF1=2
      IF(XSTAR.GT..001*XB) GO TO 1
      NLIF1=1
      WRITE(6,208)
  208 FORMAT(1H1,18H CONVERGED RESULTS//)
      OMBAR=ALD(1)
      CALL OUTPUT(ND,FSTAR)
      ALD(1)=ALD(1)+DOM
      OMEGA=ALD(1)/AMU2
      DO 104 I=1,ND
  104 XOLD(I)=XX(I)
      IF(ALD(1).GT.ALD(3)) GO TO 2
      CALL GRAD(FSTAR,1.,0.)
      CALL OUTPUT(ND,FSTAR)
      GO TO 1
  303 IF(NREAD) 304,304,305
  305 READ(5,NM2)
      GO TO 306
  304 OMXI=O1(0.)
      WO=-2./SF*(1.-COS(OMXI   ))/OMXI
      STA=NINC
      DWO=WO/STA
      N1=NINC+1
      XX(1)=WO
      DO 322 I=2,N1
  322 XX(I)=XX(I-1)-DWO
      XX(N1+1)=XX(NINC)
      DO 326 I=1,NINC
      KU=N1+1+I
  326 XX(KU)=0.
      DO 171 I=1,ND
```

```
171 XOLD(I)=XX(I)
306 CALL GRAD(FSTAR,1.,0.)
    CALL OUTPUT(ND,FSTAR)
    NLIF1=1
    DOM=DOMO
  3 F=FSTAR
    CALL ITER(ND,XSTAR,XB)
    CALL GRAD(FSTAR,1.,0.)
    WRITE(6,210) FSTAR,F
210 FORMAT(2X7H FSTAR=E16.7,2X3H F=E16.7)
    IF(ABS((FSTAR-F)/FSTAR).LT.5.E-7)  GO TO 323
    IF(FSTAR-F) 323,323,324
323 NLIF1=2
    IF(XSTAR-.001*XB) 314,314,3
324 DO 328 I=1,ND
328 XX(I)=XOLD(I)
    WRITE(6,203)
    LOC=1
    GO TO 5
203 FORMAT(1H1,31H PROCEED TO NON LINEAR SOLUTION//)
314 WRITE(6,208)
    CALL OUTPUT(ND,FSTAR)
    NLIF1=1
    DOM=-DOMO
    ALD(1)=ALD(1)+DOM
    OMEGA=ALD(1)/AMU2
    DO 325 I=1,ND
325 XX(I)=XOLD(I)
    CALL GRAD(FSTAR,1.,0.)
    CALL OUTPUT(ND,FSTAR)
    NF=1
  4 F=FSTAR
    CALL ITER(ND,XSTAR,XB)
    CALL GRAD(FSTAR,1.,0.)
    WRITE(6,210) FSTAR,F
    IF(ABS((FSTAR-F)/FSTAR).LT.5.E-4)  GO TO 311
    IF(FSTAR-F)311,311,310
311 NLIF1=2
    IF(XSTAR-0.001*XB) 312,312,4
312 WRITF(6,208)
    CALL OUTPUT(ND,FSTAR)
    NLIF1=1
    ALD(1)=ALD(1)+DOM
    OMEGA=ALD(1)/AMU2
313 DO 107 I=1,ND
107 XOLD(I)=XX(I)
327 CALL GRAD(FSTAR,1.,0.)
    CALL OUTPUT(ND,FSTAR)
    GO TO 4
310 IF(ABS(DOM)-DELTQ*DOMO) 2,2,316
316 ALD(1)=ALD(1)-.8*DOM
    DOM=DOM*.2
    OMEGA=ALD(1)/AMU2
    GO TO 4
    GO TO 313
  5 NCY=NNL*ND
    DO 6 I=1,ND
  6 XX(I)=XBAR(I)
 41 CALL GRAD(F,1.,0.)
    I1=0
 52 K1=KOUNT+6*ND
    KNL=0
    KNT=0
    OLDF=F
    DO 68 I=1,ND
 68 P(I)=-GG(I)
    IF(I1.EQ.0) CALL OUTPUT(ND,F)
    I1=1
 17 IF(KOUNT.GT.K1) GO TO 41
    CALL CYCLE(F,1.,0.,GNORM,EST,ND,AK)
    KOUNT=KOUNT+1
    KNL=KNL+1
    KNT=KNT+1
    IF(IPLUS -1) 34,25,53
 53 IF(KNT.EQ.1) RETURN
    GOTO 52
 25 WRITE(6,27)
 27 FORMAT(1H0,9H GO TO 52)
    GO TO 52
 34 IF(KNT-IPF) 17,18,18
 18 KNT=0
    IF( ABS((OLDF-F)/F).LE.ERR) GOTO 334
    OLDF=F
    WRITE(6,209) KOUNT
209 FORMAT(1H0,7H KOUNT=I5)
    CALL OUTPUT(ND,F)
    IF(NCY) 17,17,29
 29 IF(KNL-NCY) 17,17,334
334 FSTAR=F
    DO 173 I=1,ND
173 XOLD(I)=XX(I)
    GO TO 3
    END
```

```
      SUBROUTINE SEARCH(ND,EST,F,BNL,BPB,AK)
      COMMON
     1/BL4/XX(633)
     2/BL7/P(633)
     3/BL8/GG(633)
     4/BK1/X(633)
     5/BK2/IPLUS
     6/BK3/IPIF1,NCL,DELTA
CNCL IS THE NUMBER OF ITERATIONS THROUGH 3RD ORDER
C IPIF1 CONTROLS WRITTING OF INTERMEDIATE RESULTS
      DIMENSION G(633)
      EQUIVALENCE (G(1),GG(1))
      AK=0.
      FO=F
      FP=F
      YB=F
      VB=0.
      PP=0.
      DO 310 I=1, ND
      VB=VB+G(I)*P(I)
  310 PP=PP+P(I)*P(I)
      IF(VB.GT.0.) IPLUS=2
      IF(IPLUS-1) 4,4,25
   25 WRITE(6,3)
    3 FORMAT(1H1,38H ONE DIMENSIONAL SEARCH FAILURE RETURN//)
      RETURN
    4 IF(EST.LT.F) GO TO 315
      EST=F/1.5
      IF(F.LT.0.) EST=1.5*F
  315 AK=2.*(EST-F)/VB
      H=SQRT(PP)
      IF((AK.GT.0.).AND.(AK.LT.H)) H=AK
      AK=H
  320 YA=YB
      VA=VB
      IF(IPIF1.EQ.1)
     1WRITE(6,34) H,YA,VA
   34 FORMAT(
     14H  H=E16.8/
     24H YA=E16.8/
     34H VA=E16.8)
      DO 15 I=1,ND
   15 X(I)=XX(I)
   36 DO330 I=1,ND
  330 XX(I)=X (I)+H*P(I)
      CALL GRAD(F,BNL,BPB)
      YB=F
      VB=0.
      DO 340 I=1, ND
  340 VB=VB+G(I)*P(I)
      IF(IPIF1.EQ.1)
     1WRITE(6,33)H,VB
   33 FORMAT(
     14H  H=E16.8/
     24H VB=E16.8)
      IF(VB)28,400,348
   28 IF(F-YA) 11,12,12
   11 H=2.*H
      FP=F
      GO TO 320
```

```
 12 H=H+2.*H
    GO TO 36
348 K=0
 16 Z=3.0*( YA-YB )/H+VA+VB
    W1=SQRT(Z**2-VA*VB)
    BK=H*(VB+W1-Z)/(VB-VA+2.*W1)
    DO 360 I=1, ND
360 XX(I)=X (I)+(H-BK)*P(I)
    CALL GRAD(F,BNL,BPB)
    VC=0.
    P1=0.
    G1=0.
    DO 370 I=1,ND
    VC=VC+GG(I)*P(I)
    G1=G1+GG(I)*GG(I)
370 P1=P1+P(I)*P(I)
    G1=SQRT(G1)
    P1=SQRT(P1)
    PHI=ABS(VC)/P1/G1
     IF(IPIF1.EQ.1)
   1WRITE(6,385) F,YA,YB,VA,VB,BK,H,FP,VC,PHI
385 FORMAT(1H0,3H F=E16.8/4H YA=E16.8/4H YB=E16.8/4H VA=E16.8/4H VB=E1
   16.8/4H BK=E16.8/3H H=E16.8/4H FP=E16.8/
   24H VC=E16.8/5H PHI=E16.8)
     IF(F.GT.YA.OR.F.GT.YB) GO TO 29
    K=K+1
      IF (K.GE.NCL) GO TO 400
     IF(PHI.LE.DELTA) GO TO 400
     IF(F.EQ.YA.AND.YA.EQ.YB) GO TO 400
     GO TO 68
 29  IF(K) 105,105,400
105  IF(YB-YA) 37,37,60
 60  IF(F-YA)68,68,67
 68  IF(VC) 69,69,70
 69  YA=F
     H=BK
     VA=VC
     DO 76 I=1,ND
 76  X(I)=XX(I)
     GO TO 16
 70  H=H-BK
     YB=F
     VB=VC
     GO TO 16
 67  IF(VC) 38,38,39
 39  H=H-BK
     YB=F
     VB=VC
     GO TO 16
 38  H=H-BK
 40  H=H/2.
      IF(H) 46,46,47
 46  IF(IPLUS-2) 72,72,73
 72  IPLUS=1
     WRITE(6,103)
103  FORMAT(1H0,10H SEARCH 72)
     RETURN
 73  IPLUS=2
     WRITE(6,104)
104  FORMAT(1H0,10H SEARCH 73)
```

```
      RETURN
   47 CALL XVEC(H,1.,ND)
      CALL GRAD(FSTAR,BNL,BPB)
      VC=0.
      DO 41 I=1,ND
   41 VC=VC+GG(I)*P(I)
      IF(VC) 42,42,43
   42 IF(FSTAR-YA) 44,40,40
   44 DO 45 I=1,ND
   45 X(I)=XX(I)
      YA=FSTAR
      VA=VC
      GO TO 40
   43 YB=FSTAR
      VB=VC
      GO TO 16
   37 IF(F-YA) 100,100,50
  100 IF(VC) 69,69,70
   50 CALL XVEC(H,1.,ND)
      DO 48 I=1,ND
   48 X(I)=XX(I)
      H=-BK
   51 H=H/2.
      IF(H) 52,52,53
   52 IF(IPLUS-2) 74,74,75
   74 IPLUS=1
      WRITE(6,102)
  102 FORMAT(1H0,10H SEARCH 74)
      RETURN
   75 IPLUS=2
      WRITE(6,101)
  101 FORMAT(1H0,10H SEARCH 75)
      RETURN
   53 CALL XVEC(H,1.,ND)
      CALL GRAD(FSTAR,BNL,BPB)
      VC=0.
      DO 54 I=1,ND
   54 VC=VC+GG(I)*P(I)
      IF(VC) 62,62,56
   62 IF(FSTAR-YA) 55,51,51
   55 VA=VC
      YA=FSTAR
      H=-H
      DO 57 I=1,ND
   57 X(I)=XX(I)
      GO TO 16
   56 IF(FSTAR-YB) 58,51,51
   58 DO 59 I=1,ND
   59 X(I)=XX(I)
      YB=FSTAR
      VB=VC
      GO TO 51
  400 IPLUS=0
      RETURN
      END
      SUBROUTINE CYCLE (F,BNL,BPB,GNORM,EST,N,AK)
      COMMON
     1/BL7/P(633)
     2/BL8/GG(633)
     3/BK2/IPLUS
      G1=0.
      DO 1 I=1,N
    1 G1=G1+GG(I)*GG(I)
      OLDGG=G1
      CALL SEARCH(N,EST,F,BNL,BPB,AK)
      IF(IPLUS.NE.0) RETURN
      G1=0.
      DO 3 I=1,N
    3 G1=G1+GG(I)*GG(I)
      BETA=G1/OLDGG
      DO 4 I=1,N
    4 P(I)=-GG(I)+BETA*P(I)
      GNORM=SQRT(G1)
      RETURN
      END

      SUBROUTINE ITER(ND,XSTAR,XB)
      COMMON/BL4/XX(633)/BL8/GG(633)/BLK1/XBAR(633)/BLK2/GBAR(633)
      DO 5 I=1,ND
      XBAR(I)=XX(I)
    5 GBAR(I)=GG(I)
    4 CALL HGS
      XSTAR=0.
      XB=0.
      DO 6 I=1,ND
      XSTAR=XSTAR+XX(I)*XX(I)
      XX(I)=XX(I)+XBAR(I)
    6 XB=XB+XBAR(I)*XBAR(I)
      XSTAR=SQRT(XSTAR)
      XB=SQRT(XB)
      RETURN
      END

      SUBROUTINE OUTPUT(ND,FSTAR)
      COMMON/BL4/XX(633)/BL8/GG(633)/BL9/DDC(3)/BL2/Q(4)/BL18/ALD(4)
      EQUIVALENCE (PBAR ,ALD(1))
      G1=0.
      DO 1 I=1,ND
    1 G1=G1+GG(I)*GG(I)
      G1=SQRT(G1)
      WRITE(6,4) FSTAR,G1,PBAR ,DDC,Q
    4 FORMAT(1H0,7H FSTAR=E16.8,7H GNORM=E16.8// 7H PBAR =E16.8//
     218H BENDING STIFFNESS//
     1 3E16.8 //20H BOUNDARY RESTRAINTS// 4E16.8 )
      WRITE(6,2)
    2 FORMAT(1H0,14H DISPLACEMENTS//)
      WRITE(6,3)(XX(I),I=1,ND)
    3 FORMAT(8E16.8)
      WRITE(6,5)
    5 FORMAT(1H0,9H GRADIENT//)
      WRITE(6,3) (GG(I),I=1,ND)
      RETURN
      END
      SUBROUTINE XVEC(B,D,N)
      COMMON/BK1/X(633)/BL4/XX(633)/BL7/P(633)
      H=B*D
      DO 1  I=1,N
    1 XX(I)=X(I)+H*P(I)
      RETURN
      END


      FUNCTION WOGR(IND1,I,EPH,BNL,BPB)
      COMMON
     1/BL15/W(21,10,3)/BL17/ANU,SF,OMEGA
     2/BL16/NINC,NFP1
     4/BLK1/XBAR(633)
      IF(EPH.NE.0.)GO TO 3
      WOGR=0.
      RETURN
    3 IF(IND1.GT.0) GO TO 1
      K=I
      DI=FLOAT(NINC)
      RH=(FLOAT(I-1)-.5)/DI
      GO TO 2
    1 K=I+1
      DI=-FLOAT(NINC)
      RH=(FLOAT(I-1)+.5)/FLOAT(NINC)
    2 BH=B(RH)
      OXH=O1(RH)
      OTH=O2(RH)
      DH=D(RH)
      OMXT=OXH+ANU*OTH
      WOGR=(BH* OMXT*FDH(K,1,2)+DH*2.*FDH(K,1,1)*DI/RH**2)*EPH*RH
      IF(BNL.EQ.0.)GO TO 4
      WOGR=WOGR+BH*EPH*RH*SF*(SF*FDH(K,1,1)**3*DI+2.*FDH(K,1,1)*DI*(
     1OMXT*AVE(K,1,1)+FDH(K,1,2)+ANU/RH*AVE(K,1,2))+FDH(K,1,1)**2/2.
     2*OMXT)
      RETURN
    4 IF(BPB.EQ.0.)RETURN
      WA=.5*(XBAR(K)+XBAR(K-1))
      DWH=(XBAR(K)-XBAR(K-1))*FLOAT(NINC)
      KU=NINC+1+K
      IF(K-2) 6,6,5
    6 UA=.5*XBAR(KU)
      DUH=XBAR(KU)*FLOAT(NINC)
      GO TO 7
    5 UA=.5*(XBAR(KU)+XBAR(KU-1))
      DUH=(XBAR(KU)-XBAR(KU-1))*FLOAT(NINC)
    7 WOGR=WOGR+BH*RH*EPH*SF*(3.*SF*DWH**2*FDH(K,1,1)*DI+2.*FDH(K,1,1)
     1*DI*(OMXT*WA+DUH+ANU*UA/RH)+2.*DWH*DI*(OMXT*AVE(K,1,1)+FDH(K,1,2)
     2+ANU*AVE(K,1,2)/RH)+OMXT*DWH*FDH(K,1,1))
      RETURN
      END
      FUNCTION UOGR(IND1,I,EPH,BNL,BPB)
      COMMON
     1/BL15/W(21,10,3)
     2/BL16/NINC,NEP1
     3/BL17/ANU,SF,OMEGA
     4/BLK1/XBAR(633)
      IF(EPH.NE.0.)GO TO 1
      UOGR=0.
      RETURN
    1 IF(IND1.GT.0)GO TO 3
      DI=FLOAT(NINC)
      RH=(FLOAT(I-1)-.5)/DI
      K=I
      GO TO 4
    3 DI=-FLOAT(NINC)
      RH=(FLOAT(I-1)+.5)/FLOAT(NINC)
      K=I+1
    4 BH=B(RH)
      DH=D(RH)
      OX=O1(RH)
      OT=O2(RH)
      O12=OX+ANU*OT
      O21=OT+ANU*OX
      UOGR=BH*EPH*RH*(2.*FDH(K,1,2)*DI+2.*O12*AVE(K,1,1)*DI+ANU/RH*FDH(
     1K,1,2)+2.*ANU/RH*AVE(K,1,2)*DI)
      IF(BNL.EQ.0.)GO TO 5
      UOGR=UOGR+BH*EPH*RH*(FDH(K,1,1)**2*DI+ANU/RH*.5*FDH(K,1,1)**2)*SF
    5 IF(BPB.EQ.0.)RETURN
      UOGR=UOGR+BH*EPH*RH*SF*2.*(XBAR(K)-XBAR(K-1))*FLOAT(NINC)*FDH(K,
     11,1)*(DI+ANU/2./RH)
      RETURN
      END


      FUNCTION WOIGR(I,EP)
      COMMON
     1/BL15/W(21,10,3)
     2/BL16/NINC,NEP1
     3/BL17/ANU,SF,OMEGA
      IF(EP.NE.0.)GO TO 1
      WOIGR=0.
      RETURN
    1 DEL=1./FLOAT(NINC)
      D2=1./DEL**2
      RO=FLOAT(I-1)*DEL
      OX=O1(RO)
      OT=O2(RO)
      BI=B(RO)
      DI=D(RO)
      WOIGR=(BI*(2.*W(I,1,1)*(OX**2+OT**2+2.*ANU
     2                                          *OX*OT)+2./RO*(OT+ANU*OX)*
     1W(I,1,2))+DI*(-4.*D2*SD(I,1)-4.*ANU/RO*FDI(I,1,1)*D2))*EP*RO
      RETURN
      END
      FUNCTION WOMPG(IND1,I,EP)
      COMMON
     1/BL15/W(21,10,3)
     2/BL16/NINC,NEP1
     3/BL17/ANU,SE,OMEGA
      IF(EP.NE.0.) GO TO 1
      WOMPG=0.
      RETURN
    1 DEL=1./FLOAT(NINC)
      D2=1./DEL**2
      IF(IND1.GT.0)GO TO 2
      RO=FLOAT(I-2)*DEL
      K=I-1
      J=I
      GO TO 3
    2 RO=FLOAT(I)*DEL
      K=I+1
      J=I+1
    3 DI=D(RO)
      WOMPG=DI*EP*(2.*SD(K,1)*D2*RO+2.*ANU*FDH(J,1,1)*D2)
      RETURN
      END
```

```
      FUNCTION UOI(I,EP)
      COMMON
     1/BL15/W(21,10,3)
     2/BL16/NINC,NEP1
     3/BL17/ANU,SE,OMEGA
      IF(EP.NE.0.)GO TO 1
      UOI=0.
      RETURN
    1 DEL=1./FLOAT(NINC)
      RO=FLOAT(I-1)*DEL
      BI=B(RO)
      UOI=BI*EP*RO*(2.*W(I,1,2)/RO**2+2./RO*(O2(RO)+ANU*O1(RO))*W(I,1,1)
     1)
      RETURN
      END
      SUBROUTINE MOVEX
      COMMON
     1/BL15/W(21,10,3)
     2/BL4/XX(633)
     3/BL16/NINC,NFP1
      N1=NINC+1
      N2=NINC+2
      W(1,1,2)=0.
      W(1,1,1)=XX(1)
      NW=NINC+2+(NFP1-1)*(NINC+1)
      NU=NINC
      IF(NFP1.GT.1)NU=NINC+(NFP1-1)*NINC+1
      NV=0
      IF(NFP1.GT.1)NV=(NFP1-1)*NINC
      DO 1 I=2,N2
    1 W(I,1,1)=XX(I)
      DO 4 I=2,N1
      K=NW-1+I
    4 W(I,1,2)=XX(K)
      IF(NFP1-2)2,3,3
    3 KU=NW+NINC+1
      W(1,2,2)=XX(KU)
      DO 5 N=2,NFP1
      DO 5 I=2,N2
      KW=NINC+1+(N-2)*(NINC+1)+I
    5 W(I,N,1)=XX(KW)
      DO 6 N=2,NFP1
      DO 6 I=2,N1
      KU=NW+NINC+I+(N-2)*NINC
      KV=NW+NU-1+I+(N-2)*NINC
      W(I,N,2)=XX(KU)
    6 W(I,N,3)=XX(KV)
    2 RETURN
      END
      SUBROUTINE GRAD(E,BNL,BPB)
      COMMON
     1/BL8/GG(633)
     2/BL16/NINC,NFP1
     3/BLK2/GBAR(633)/BLK3/PIBAR
      CALL MOVEX
      DX=1./FLOAT(NINC)
      N1=NINC+1
      N2=NINC+2
      DO 13 K=1,2
      M=1
      N=N2
      IF(K-1) 15,15,14
   14 M=2
      N=N1
   15 DO 12 I=M,N
      E=DX
      EM=DX
      EP=DX
      EMH=DX
      EPH=DX
      IF(I-2)1,2,3
    1 E=0.
      EM=0.
      EMH=0.
      GO TO 4
    2 EM=0.
    3 IF(I.EQ.NINC)EP=.5*DX
      IF(I-N1) 4,5,6
    5 E=.5*DX
      EP=0.
      EPH=0.
      GO TO 4
    6 EM=.5*DX
      E=0.
      EP=0.
      EPH=0.
      EMH=0.
    4 IF(K-1)11,11,12
   11 GG(I)=WOGR(1,I,EPH,BNL,BPB)+WOGR(-1,I,EMH,BNL,BPB)+WOIGR(I,E)+
     1WOMPG(1,I,EP)+WOMPG(-1,I,EM)
      GO TO 13
   12 KU=N1+I
      GG(KU)=UOGR(1,I,EPH,BNL,BPB)+UOGR(-1,I,EMH,BNL,BPB)+UOI(I,E)
   13 CONTINUE
      CALL EDGEGR
      IF(BPB.EQ.0.) GO TO 16
      NM=2*NINC+2
      DO 9 I=1,NM
    9 GG(I)=GG(I)+GBAR(I)
      RETURN
   16 CALL LOADGR
      CALL POTEN(E,BNL,BPB)
      RETURN
      END
      SUBROUTINE POTEN(F,BNL,BPB)
C THE INDICATORS BNL AND BPB HAVE THE FOLLOWING SIGNIFICANCE
C BNL=0. SIGNIFIES THAT NON LINEAR TERMS ARE EXCLUDED
C BNL=1. SIGNIFIES THAT NON LINEAR TERMS ARE INCLUDED
C BPB=0. SIGNIFIES THAT PERTURBATION TERMS ARE EXCLUDED
C BPB=1. SIGNIFIES THAT PERTURBATION TERMS ARE INCLUDED
      COMMON
     1/BL15/W(21,10,3)
     2/BL16/NINC,NEP1
     3/BL17/ANU,SF,OMEGA
     4/BLK1/XBAR(633)
     5/BLK3/PIBAR
     6/BLK2/GBAR(633)
     7/BL4/XX(633)
      V1=0.
      V2=0.
      V3=0.
      V4=0.
      DEL=1./FLOAT(NINC)
      N1=NINC+1
      DO 1 I=2,N1
      EP=DEL
      RO=FLOAT(I-1)*DEL
      RMH=RO-.5*DEL
      DH=D(RMH)
      EMH=DEL
      IF(I.EQ.N1)EP=.5*DEL
      BH=B(RMH)
      BI=B(RO)
      OX=O1(RO)
      OT=O2(RO)
      OXM=O1(RMH)
      OTM=O2(RMH)
      DI=D(RO)
      OXT=OXM+ANU*OTM
      OTX=OTM+ANU*OXM
      V1=V1+BH*EMH*RMH*(FDH(I,1,2)**2+2.*OXT*AVE(I,1,1)*FDH(I,1,2)+2.*
     1ANU*AVE(I,1,2)*FDH(I,1,2)/RMH)+EMH*DH*FDH(I,1,1)**2/RMH
     2+BI*EP*RO*(W(I,1,1)**2*(OX**2+OT**2+2.*ANU*OT*OX)+(W(I,1,2)/RO)**2
     3+2.*(OT+ANU*OX)*W(I,1,1)*W(I,1,2)/RO)+DI*EP*RO*(SD(I,1)**2+2.*ANU
     4*FDI(I,1,1)*SD(I,1)/RO)
      IF(BPB.NE.0.) GO TO 2
      IF(BNL.EQ.0.) GO TO 1
      V2=V2+BH*EMH*RMH*SF*(SF*FDH(I,1,1)**4/4.+(AVE(I,1,1)*OXT+FDH(I,1,
     12)+ANU*AVE(I,1,2)/RMH)*FDH(I,1,1)**2)
      GO TO 1
    2 WA=(XBAR(I)+XBAR(I-1))*.5
      DWH=(XBAR(I)-XBAR(I-1))/DEL
      KU=N1+I
      IF(I-2) 3,3,4
    4 UA=(XBAR(KU)+XBAR(KU-1))*.5
      DU=(XBAR(KU)-XBAR(KU-1))/DEL
      GO TO 5
    3 UA=.5*XBAR(KU)
      DU=XBAR(KU)/DEL
    5 V4=V4+BH*EMH*RMH*SF*(SF/4.*(                              6.*DWH**2
     1*FDH(I,1,1)**2)+(OXT*WA+DU+ANU*UA/RMH)*(
     2 FDH(I,1,1)**2)+(OXT*AVE(I,1,1)+FDH(I,1,2)+ANU*AVE(I,1,2)/RMH)*
     3(          2.*DWH*FDH(I,1,1)))
    1 CONTINUE
```

```
  IF(BPB.EQ.0.) GO TO 7
  V3=0.
  ND=2*NINC+2
  DO 6 I=1,ND
6 V3=V3+GBAR(I)*XX(I)
7 F=V1+V2+V3+V4+PILOAD(BNL,BPB)+PIEDGE(BNL,BPB)+PIBAR
  RETURN
  END


  FUNCTION PILOAD(BNL,BPB)
  COMMON
 1/BL1/PN(10),ML
 2/BL15/W(21,10,3)
 3/BL16/NINC,NFP1
 4/BL17/ANU,SF,OMEGA
  PILOAD=0.
  IF(BPB.NE.0.) RETURN
  DEL=1./FLOAT(NINC)
  P1=0.
  P2=0.
  N1=NINC+1
  DO 1 I=2,N1
  RO=FLOAT(I-1)*DEL
  EP=DEL
  IF(I.EQ.N1   )EP=.5*DEL
  P1=P1-2.*PN(1)*W(I,1,1)*RO*EP
  IF(ML-1)1,1,2
2 DO 3 N=2,ML
3 P2=P2-PN(N)*W(I,N,1)*RO*EP
1 CONTINUE
  PILOAD=OMEGA/SF*(P1+P2)
  RETURN
  END


  FUNCTION PIEDGE(BNL,BPB)
  COMMON
 1/BL15/W(21,10,3)
 2/BL2/Q(4)
 3/BL16/NINC,NFP1
 4/BLK1/XBAR(633)
  N1=NINC+1
  PIEDGE=Q(1)*W(N1,1,2)**2+Q(3)*W(N1,1,1)**2+Q(4)*FDI(N1,1,1)**2
 2 +2.*Q(2)*W(N1,1,1)*W(N1,1,2)
  RETURN
  END
```

```
      SUBROUTINE LOADGR
      COMMON
     1/BL15/W(21,10,3)
     2/BL16/NINC,NFP1
     3/BL1/PN(10),ML
     4/BL8/GG(633)
     5/BL17/ANU,SF,OMEGA
      N1=NINC+1
      DEL=1./FLOAT(NINC)
      DO 1  I=2,N1
      E=DEL
      IF(I.EQ.N1)  E=.5*DEL
      RO=FLOAT(I-1)*DEL
      OM1=OMEGA/SF
    1 GG(I)=GG(I)-2.*OM1*PN(1)*E*RO
      IF(ML.LT.2)RETURN
      N1=NINC+1
      DO 2  N=2,ML
      DO 2  I=2,N2
      K=N2-1+I+(N-2)*N1
      E=DEL
      IF(I.EQ.N1)E=.5
      IF(I.EQ.N2)E=0.
    2 GG(K)=GG(K)-OM1    *PN(N)*E*RO
      RETURN
      END


      SUBROUTINE EDGEGR
      COMMON
     1/BL2/Q(4)
     2/BL8/GG(633)
     3/BL15/W(21,10,3)
     4/BL16/NINC,NFP1
     5/BLK1/XBAR(633)
      N1=NINC+1
      N2=NINC+2
      DEL=1./FLOAT(NINC)
      A=    Q(4)/DEL*FDI(N1,1,1)
      GG(N1)=GG(N1)+2.*Q(3)*W(N1,1,1)+2.*Q(2)*W(N1,1,2)
      GG(NINC)=GG(NINC)-A
      GG(N2)=GG(N2)+A
      NW1=N2+NINC
      GG(NW1     )=GG(NW1     )+2.*Q(1)*W(N1,1,2)+2.*Q(2)*W(N1,1,1)
      RETURN
      END


      FUNCTION FDI(I,J,K)
      COMMON/BL15/W(21,10,3)/BL16/NINC,NFP1
      W(1,2,3)=W(1,2,2)
      FDI=(W(I+1,J,K)-W(I-1,J,K))/2.*FLOAT(NINC)
      RETURN
      END
      FUNCTION SD(I,J)
      COMMON/BL15/W(21,10,3)/BL16/NINC,NFP1
      SD=(W(I+1,J,1)-2.*W(I,J,1)+W(I-1,J,1))*FLOAT(NINC)**2
      RETURN
      END
      FUNCTION FDH(I,J,K)
      COMMON/BL15/W(21,10,3)/BL16/NINC,NFP1
      W(1,2,3)=W(1,2,2)
      FDH=(W(I,J,K)-W(I-1,J,K))*FLOAT(NINC)
      RETURN
      END


      FUNCTION AVE(I,J,K)
      COMMON/BL15/W(21,10,3)/BL16/NINC,NFP1
      W(1,2,3)=W(1,2,2)
      AVE=(W(I,J,K)+W(I-1 ,J,K))/2.
      RETURN
      END


      SUBROUTINE HGS
      COMMON
     1/COEF/AMAT(14)
     2/BL17/ANU,SF,OMEGA
     3/BLK1/XBAR(633)
     4/BL16/NINC,NFP1
     5/NLCOEF/ENMAT(9)
     6/BL3/QXC(3)
     7/BL5/QTC(3)
     8/BL6/BBC(3)
     9/BL9/DDC(3)
     A/BL2/Q(4)
     B/BL4/XX(633)
     C/NONLIN/WN,UN,FNL
     D/BLK2/GBAR(633)
     E/BK3/IPIF1,NCL,DELTA
      DIMENSION A(22,22),AL(22),IPIVOT(22)
    4 FORMAT(8E16.8)
      BNL=0.
      BPB=0.
      DO 38 I=1,22
      DO 38 J=1,22
   38 A(I,J)=0.
      IF(IPIF1.EQ.1) WRITE(6,2)
    2 FORMAT(1H1,27H MATRIX COEFFICIENTS FOLLOW//)
      DO 16 I=1,9
   16 ENMAT(I)=0.
   18 DO 1 I=1,12
      CALL LMATC(I,BNL,BPB)
      A(I,I)=AMAT(3)+A(I,I)+ENMAT(2)
      IF(I-2) 7,7,9
    9 IF(I-11)7,10,14
    7 A(I,I+2)=AMAT(5)+A(I,I+2)
   10 A(I,I+1)=AMAT(4)+A(I,I+1)+ENMAT(3)
   14 IF(I-2) 11,12,13
   13 A(I,I-2)=AMAT(1)+A(I,I-2)
   12 A(I,I-1)=AMAT(2)+A(I,I-1)+ENMAT(1)
   11 IF(IPIF1.NE.1) GO TO 1
      WRITE(6,3) I
    3 FORMAT(1H0,15H STATION NUMBERI5//)
      IF(BPB.EQ.0.) WRITE(6,4) AMAT
      IF(BPB.NE.0.) WRITE(6,4) ENMAT
    1 CONTINUE
      DO 17 I=1,12
      K=I+11
      CALL LMATC(I,BNL,BPB)
      IF(I-2) 24,22,25
   24 A(I,K+1)=A(I,K+1)+AMAT(8)+ENMAT(6)
      GO TO 34
   25 IF(I-11) 22,22,26
   26 A(I,K-1)=A(I,K-1)+AMAT(6)+ENMAT(4)
      GO TO 34
   22 A(I,K)=A(I,K)+AMAT(7)+ENMAT(5)
      A(K,I+1)=A(K,I+1)+AMAT(14)+ENMAT(9)
      A(K,I-1)=A(K,I-1)+AMAT(12)+ENMAT(7)
      A(K,I)=A(K,I)+AMAT(13)+ENMAT(8)
      A(K,K)=A(K,K)+AMAT(10)
      IF(I-2)23,23,27
   27 A(K,K-1)=A(K,K-1)+AMAT(9)
      A(I,K-1)=A(I,K-1)+AMAT(6)+ENMAT(4)
      IF(I-11) 23,23,34
   23 A(I,K+1)=A(I,K+1)+AMAT(8)+ENMAT(6)
      A(K,K+1)=A(K,K+1)+AMAT(11)
   34 IF(IPIF1.NE.1) GO TO 17
      WRITE(6,3) I
      IF(BPB.EQ.0.) WRITE(6,4) AMAT
      IF(BPB.EQ.1.) WRITE(6,4) ENMAT
   17 CONTINUE
      IF(BPB.NE.0.) GO TO 28
      DO 33 I=1,14
   33 AMAT(I)=0.
      BPB=1.
      GO TO 18
   28 DO 15 I=1,22
   15 AL(I)=-GBAR(I)
      IF(IPIF1.NE.1) GO TO 39
      WRITE(6,30)
   30 FORMAT(1H0,16H MATRIX ELEMENTS//)
      DO 35 I=1,22
      WRITE(6,36) I
   36 FORMAT(1H0,8H ROW NO I5//)
   35 WRITE(6,4) (A(I,J),J=1,22)
      WRITE(6,31)
   31 FORMAT(1H0,13H FORCE VECTOR//)
      WRITE(6,4) AL
   39 CALL SIMEQ(A,22,AL,1,DET,IPIVOT,22,IS)
      IF(IPIF1.NE.1) GO TO 101
      WRITE(6,32)
   32 FORMAT(1H0,9H SOLUTION//)
      WRITE(6,4) AL
  101 DO 37 I=1,22
   37 XY(I)=AL(I)
      RETURN
      END
      SUBROUTINE  LMATC(I.BNL.BPB)
      COMMON
     1/COEF/AMAT(14)
     2/BL16/NINC,NFP1
     3/NLCOEF/ENMAT(9)
     4/BL2/Q(4)
     5/NONLIN/WN,UN,FNL
      DEL=1./FLOAT(NINC)
      E=DEL
      EP=DEL
      EM=DEL
      EPH=DEL
      EMH=DEL
      IF(I-2)1,2,3
    1 EM=0.
      EMH=0.
      E=0.
      GO TO 3
    2 EM=0.
    3 N1=NINC+1
      IF(I-N1)4,5,6
    5 E=.5*DEL
      EP=0.
      EPH=0.
      GO TO 4
    6 EM=.5*DEL
      EMH=0.
      EPH=0.
      E=0.
      EP=0.
    4 RO=DEL*FLOAT(I-1)
      IF(I.EQ.NINC) EP=.5*DEL
      RP=RO+DEL
      RM=RO-DEL
      RPH=RO+DEL/2.
      RMH=RO-DEL/2.
      BE=B(RO)
      DE=D(RO)
      BMH=B(RMH)
      BPH=B(RPH)
      DMH=D(RMH)
      DPH=D(RPH)
      DP=D(RP)
      DM=D(RM)
      OX=O1(RO)
      OT=O2(RO)
      OXP=O1(RPH)
      OTP=O2(RPH)
      OXM=O1(RMH)
      OTM=O2(RMH)
      IF(BNL.NE.0..OR.BPB.NE.0.) GO TO 12
      DO 7 J=1,14
    7 AMAT(J)=0.
      CALL AM(EM,RM,DM,DEL)
      CALL AP(EP,RP,DP,DEL)
      CALL AE(E,RO,DE,BE,OT,OX,DEL)
      CALL AMH(EMH,RMH,DMH,BMH,OTM,OXM,DEL)
      CALL APH(EPH,RPH,DPH,BPH,OTP,OXP,DEL)
      A=0.
      T=0.
      C=0.
      IF(I.EQ.N1) A=1.
      IF(I.EQ.NINC) C=1.
      IF(I.EQ.(NINC+2)) T=1.
      AMAT(3)=AMAT(3)+2.*Q(3)*A+    Q(4)*(T+C)/2./DEL/DEL
      AMAT(1)=AMAT(1)-Q(4)/DEL*T/2./DEL
      AMAT(5)=AMAT(5)-Q(4)/DEL*C/2./DEL
      AMAT(7)=AMAT(7)+2.*Q(2)*A
      AMAT(13)=AMAT(13)+2.*Q(2)*A
      AMAT(10)=AMAT(10)+Q(1)*2.*A
      RETURN
   12 IF(BNL.EQ.1.) GO TO 8
      DO 11 J=1,9
   11 ENMAT(J)=0.
      K=I+1
      CALL NMH(FMH,RMH,BMH,OTM,OXM,DEL,I,BNL,BPB)
      CALL NPH(FPH,RPH,BPH,OTP,OXP,DEL,K,BNL,BPB)
    8 RETURN
      END


      SUBROUTINE AM(E,RO,D,DEL)
      COMMON
     1/COEF/AMAT(14)
     2/BL17/ANU,SF,OMEGA
      IF( E.EQ.0.)RETURN
      ERD=2.*E*RO*D
      DEL2=DEL**2
      DEL4=DEL2*DEL2
      AMAT(1)=ERD/DEL2**2+AMAT(1)
      AMAT(2)=-ERD*(2./DEL4+ANU/RO/DEL**3)+AMAT(2)
      AMAT(3)=ERD*(1./DEL4+ANU/RO/DEL**3)+AMAT(3)
      RETURN
      END


      SUBROUTINE AP(E,RO,D,DEL)
      COMMON
     1/COEF/AMAT(14)
     2/BL17/ANU,SF,OMEGA
      IF(E.EQ.0.)RETURN
      ERD=2.*E*RO*D
      DEL2=DEL**2
      DEL4=DEL2*DEL2
      AMAT(3)=AMAT(3)+ERD*(1./DEL4-ANU/RO/DEL**3)
      AMAT(4)=AMAT(4)+ERD*(-2./DEL4+ANU/RO/DEL**3)
      AMAT(5)=AMAT(5)+ERD/DEL**4
      RETURN
      END
      SUBROUTINE AE(E,RO,D,B,OT,OX,DEL )
      COMMON
     1/COEF/AMAT(14)
     2/BL17/ANU,SF,OMEGA
      IF(E.EQ.0.)RETURN
      ERD=E*RO*D*2.
      DEL2=DEL*DEL
      DEL3=DEL2*DEL
      DEL4=DEL3*DEL
      AMAT(2)=AMAT(2)+ERD*(-2./DEL4+ANU/RO/DEL3)
      AMAT(3)=AMAT(3)+E*RO*B*2.*(OX**2+OT**2+2.*ANU*OX*OT)+ERD*4./DEL4
      AMAT(4)=AMAT(4)+ERD*(-2./DEL4-ANU/RO/DEL3)
      AMAT(7)=AMAT(7)+E*B*2.*(OT+ANU*OX)
      AMAT(10)=AMAT(10)+2.*E*B/RO
      AMAT(13)=AMAT(13)+2.*E*B*(OT+ANU*OX)
      RETURN
      END



      SUBROUTINE AMH(E,RO,D,B,OT,OX,DEL)
      COMMON
     1/COEF/AMAT(14)
     2/BL17/ANU,SF,OMEGA
      IF(E.EQ.0.) RETURN
      OXT=OX+ANU*OT
      ERB=E*RO*B*OXT/DEL
      AMAT(2)=AMAT(2)-E/DEL**2*D/RO*2.
      AMAT(3)=AMAT(3)+2.*E/DEL**2*D/RO
      AMAT(6)=AMAT(6)-ERB
      AMAT(7)=AMAT(7)+ERB
      AMAT(9)=AMAT(9)-2./DEL**2*E*RO*B
      AMAT(10)=AMAT(10)+E*RO*B*(2./DEL**2+2.*ANU/RO/DEL)
      AMAT(12)=AMAT(12)+ERB
      AMAT(13)=AMAT(13)+ERB
      RETURN
      END



      SUBROUTINE APH(E,RO,D,B,OT,OX,DEL)
      COMMON
     1/COEF/AMAT(14)
     2/BL17/ANU,SF,OMEGA
      IF(E.EQ.0.)RETURN
      OXT=OX+ANU*OT
      ERB=E*RO*B*OXT/DEL
      DEL2=DEL**2
      EOD=E/DEL2*D/RO
      AMAT(3)=AMAT(3)+2.*EOD
      AMAT(4)=AMAT(4)-2.*EOD
      AMAT(7)=AMAT(7)-ERB
      AMAT(8)=AMAT(8)+ERB
      AMAT(10)=AMAT(10)+E*RO*B*(2./DEL2-2.*ANU/RO/DEL)
      AMAT(11)=AMAT(11) -2./DEL2*E*RO*B
      AMAT(13)=AMAT(13)-ERB
      AMAT(14)=AMAT(14)-ERB
      RETURN
      END
      SUBROUTINE NMH(EMH,RMH,BMH,OTM,OXM,DEL,I,BNL,BPB)
      COMMON
     1/NLCOEF/ENMAT(9)
     2/BL17/ANU,SF,OMEGA
     3/BLK1/XBAR(633)/BL16/NINC,NFP1
     4/BL4/XX(633)
      IF(EMH.EQ.0.) RETURN
      IF(BPB.EQ.0..AND.BNL.EQ.0.) RETURN
      IF(BPB.EQ.1.) GO TO 4
      N=2*NINC+2
      DO 5 M=1,N
    5 XBAR(M)=XX(M)
    4 DWH=(XBAR(I)-XBAR(I-1))/DEL
      WA=(XBAR(I)+XBAR(I-1))*.5
      KU=NINC+1+I
      IF(I-2) 1,1,2
    1 DUH=XBAR(KU)/DEL
      UA=.5*XBAR(KU)
      GO TO 3
    2 DUH=(XBAR(KU)-XBAR(KU-1))/DEL
      UA=(XBAR(KU)+XBAR(KU-1))*.5
    3 BRE=BMH*RMH*EMH*SF
      D2=DEL**2
      O12=(OXM+ANU*OTM)
      ENMAT(1)=ENMAT(1)+BRE*(-3.*SF*DWH**2/D2-2./D2*(WA*O12+DUH+ANU/RMH
     1*UA))
      ENMAT(2)=BRE*(3.*SF*DWH**2/D2+DWH*O12/DEL*2.+2./D2*(WA*O12+DUH
     1+ANU/RMH*UA))+ENMAT(2)
      A=2.*DWH/DEL*BRE
      T=(-1./DEL+ANU/2./RMH)
      C=(1./DEL+ANU/2./RMH)
      ENMAT(4)=ENMAT(4)+A*T
      ENMAT(5)=ENMAT(5)+A*C
      ENMAT(7)=ENMAT(7)-A*C
      ENMAT(8)=ENMAT(8)+A*C
      RETURN
      END


      SUBROUTINE NPH(EPH,RPH,BPH,OTP,OXP,DEL,K,BNL,BPB)
      COMMON
     1/NLCOEF/ENMAT(9)
     2/BL17/ANU,SF,OMEGA
     3/BL16/NINC,NFP1
     4/BLK1/XBAR(633)
     5/BL4/XX(633)
      IF(EPH.EQ.0.)RETURN
      IF(BNL.EQ.0..AND.BPB.EQ.0.) RETURN
      IF(BPB.EQ.1.) GO TO 4
      N=2*NINC+2
      DO 5 M=1,N
    5 XBAR(M)=XX(M)
    4 DWH=(XBAR(K)-XBAR(K-1))/DEL
      WA=(XBAR(K)+XBAR(K-1))*.5
      KU=NINC+1+K
      IF(K-2)1,1,2
    1 DUH=XBAR(KU)/DEL
      UA=.5*XBAR(KU)
      GO TO 3
    2 DUH=(XBAR(KU)-XBAR(KU-1))/DEL
```

```
      UA=.5*(XBAR(KU)+XBAR(KU-1))
3     BRE=RPH*RPH*RPH*SF
      D2=DEL**2
      O12=(OXP+ANU*OTP)
      ENMAT(2)=ENMAT(2)+BRE*(3.*SF*DWH**2/D2-2.*DWH*O12/DEL+2./D2*(
     1WA*O12+DUH+ANU/RPH*UA))
      ENMAT(3)=ENMAT(3)-BRE*(3.*SF*DWH**2/D2+2./D2*(WA*O12+DUH+ANU/RPH
     1*UA))
      A=2.*DWH/DEL*BRE
      T=(-1./DEL+ANU/2./RPH)
      C=(1./DEL+ANU/2./RPH)
      ENMAT(5)=ENMAT(5)-A*T
      ENMAT(6)=ENMAT(6)-A*C
      ENMAT(8)=ENMAT(8)-A*T
      ENMAT(9)=ENMAT(9)+A*T
      RETURN
      END




      FUNCTION B(RO)
      COMMON/BL6/BBC(3)
      B=BBC(1)+BBC(2)*RO+BBC(3)*RO**2
      RETURN
      END




      FUNCTION D(RO)
      COMMON/BL9/DDC(3)
      D=DDC(1)+DDC(2)*RO+DDC(3)*RO**2
      RETURN
      END




      FUNCTION O1(RO)
      COMMON/BL3/OXC(3)
      EQUIVALENCE (OMZ,OXC(1)),(ALPHA,OXC(2))
      Z(RO,ALPHA,OMZ)=ALPHA**2/OMZ*SQRT(1.-(OMZ/ALPHA*RO)**2)
      O1=ALPHA**2/Z(RO,ALPHA,OMZ)*(1.+(ALPHA*RO/Z(RO,ALPHA,OMZ))**2)/
     1SQRT(1.+(ALPHA**2*RO/Z(RO,ALPHA,OMZ))**2)**3
      RETURN
      END




      FUNCTION O2(RO)
      COMMON/BL3/OXC(3)
      EQUIVALENCE (OMZ,OXC(1)),(ALPHA,OXC(2))
      Z(RO,ALPHA,OMZ)=ALPHA**2/OMZ*SQRT(1.-(OMZ/ALPHA*RO)**2)
      O2             =ALPHA**2/Z(RO,ALPHA,OMZ)/SQRT(1.+ALPHA**4*RO**2/
     1Z(RO,ALPHA,OMZ)**2)
      RETURN
      END
```